



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

ČASOVAČ PRO RC MODELY

TIMER FOR RC MODELS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Ondřej Fišer

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jiří Šebesta, Ph.D.

BRNO 2018

Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Bc. Ondřej Fišer

ID: 164265

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Časovač pro RC modely

POKyny PRO VYPRAcOVÁNÍ:

Prostudujte možnosti realizace časovače pro RC modely letadel. Navrhněte koncepci zařízení splňující uživatelské požadavky. Zařízení umožní činnost s regulátory s BEC, tak i použití LiPo článku, bude modulární. Navržené zařízení v závislosti na čase bude ovládat dle předem vloženého algoritmu regulátor a bude přímo ovládat další dvě doplňkové funkce – typicky servomechanismy. Navrhněte zapojení, desky plošných spojů a mechanické řešení.

Realizujte navržené zařízení, vytvořte software pro programování požadovaných funkcí časovače. Ověřte funkci zařízení a sestavte podrobný uživatelský manuál.

DOPORUČENÁ LITERATURA:

[1] BURKHARD, M. C pro mikrokontroléry. Praha: BEN - technická literatura, 2003.

[2] FRÝZA, T., FEDRA, Z., ŠEBESTA, J. Mikroprocesorová technika. Počítačová cvičení. Elektronické skriptum. Brno: FEKT VUT v Brně, 2009.

Termín zadání: 5. 2. 2018

Termín odevzdání: 17. 5. 2018

Vedoucí práce: doc. Ing. Jiří Šebesta, Ph.D.

Konzultant: Ing. Kamil Pítra, Ph.D. (VTÚO Brno)

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady



UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Tato práce se zabývá kompletním návrhem časovače pro RC modely. Popisovaný časovač má tři PWM výstupy. Pomocí těchto výstupů lze řídit dva nezávislé servomotory a regulátor motoru. Celý výrobek je řízen pomocí 8-bitového mikrokontroleru. Časové záznamy jsou uloženy v externí paměti EEPROM, kterou lze pomocí sběrnice I2C programovat. Práce se dále zabývá návrhem programovacího modulu, pomocí kterého je možné časovač pohodlně programovat přes PC nebo smartphone. Cílem práce je návrh a realizace RC časovače s příslušenstvím špičkové kvality.

KLÍČOVÁ SLOVA

RC model, časovač, MCU, servomotor, ATmega, PWM modulace

ABSTRACT

This paper describes complete design of timer for RC models. Described timer has three PWM outputs. With these outputs it is possible to connect two independent servo-motors and engine controller. 1 signal. The whole is controlled by 8-bit microprocessor. The time records are stored in external EEPROM, that can be programmed using I2C. Paper further discuss design of programming, which can be easily programmed the memory of EEPROM by the PC or smartphone. The goal of this paper is to design and build high quality RC timer with accessories.

KEYWORDS

RC model, timer, MCU, servo-motor, ATmega, PWM modulation

FIŠER, O. *Časovač pro RC modely*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2018. 45 s., 19 s. příloh. Diplomová práce. Vedoucí práce: doc. Ing. Jiří Šebesta, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Časovač pro RC modely jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce panu doc. Ing. Jiřímu Šebestovi, Ph.D. a panu Ing. Kamilu Pítrovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

.....

(podpis autora)

OBSAH

Seznam obrázků	viii
Seznam tabulek	x
Úvod	1
1 ČASOVAČ PRO RC MODELÝ	2
1.1 Servomotory.....	2
1.2 Řízení serv a regulátoru motoru.....	3
1.2.1 Analogové řízení.....	3
1.2.2 Digitální řízení	4
1.3 Regulátor otáček	5
1.4 Přijímač pro RC modely	7
1.5 Časovač pro RC modely	9
2 NÁVRH HARDWARE	10
2.1 Požadavky na návrh časovače.....	10
2.2 Výpočet a volba vhodné EEPROM paměti	11
2.3 Mikrokontroler ATmega8.....	12
2.4 Praktická realizace prototypu RC časovače.....	14
3 ŘÍDÍCÍ SOFTWARE PRO ČASOVAČ	15
3.1 Běh programu	15
3.2 Funkce programu	16
3.3 Generování výstupního PWM signálu.....	17
3.4 Hlídač délky záznamů.....	18
3.5 Časovače a přerušení	18
3.6 Scheduler	19
3.7 Odstranění zákmitů startovacího tlačítka.....	19
4 PROGRAMOVACÍ MODUL	20
4.1 Procesorová jednotka.....	21
4.2 Struktura vnitřní paměti programovacího modulu.....	22
4.3 OLED displej	23
4.4 Kapacitní tlačítka	24
4.5 Bluetooth.....	25

4.6	USB/UART převodník	26
4.7	Napájení	27
4.8	Praktická realizace prototypu programovacího modulu	28
5	ŘÍDÍCÍ SOFTWARE PROGRAMOVACÍHO MODULU	29
5.1	Běh programu	29
5.2	Funkce programu	30
5.3	Zápis dat do vnitřní paměti	31
5.4	Formátování dat pro uložení do vnitřní paměti	32
5.5	Příjem dat po sériových linkách.....	33
5.6	Ovládání firmware programovacího modulu.....	33
5.6.1	Plocha.....	34
5.6.2	Import z EEPROM.....	35
5.6.3	Nastavení	35
5.6.4	Vypnutí	35
5.6.5	Pohyb v nabídkách programovacího modulu	36
5.7	Komunikační příkazy pro programovací modul	36
6	OVLÁDACÍ APLIKACE PRO WINDOWS	37
7	OVLÁDACÍ APLIKACE PRO ANDROID	39
8	ZÁVĚR	41
	Literatura	42
	Seznam symbolů, veličin a zkratk	43
	Seznam příloh	45
	Obsah přiloženého cd	66

SEZNAM OBRÁZKŮ

Obrázek 1.1: Blokové schéma RC modelu	2
Obrázek 1.2: Řez modelářským servem	3
Obrázek 1.3: Řídící impulzy analogového serva	4
Obrázek 1.4: Řídící impulzy analogového serva	5
Obrázek 1.5: Srovnání pásma necitlivosti analogového a digitálního serva	5
Obrázek 1.6: Blokové schéma regulátoru motoru	6
Obrázek 1.7: RC přijímač od firmy Hitec.....	7
Obrázek 1.8: Blokové schéma FM/PCM přijímače s dvojím směřováním.....	8
Obrázek 1.9: Komerčně prodávaný časovač Valkyrie pro RC modely	9
Obrázek 2.1: Blokové schéma RC časovače	10
Obrázek 2.2: Schéma zapojení paměti EEPROM	11
Obrázek 2.3: Vnitřní organizace paměti záznamů časovače	12
Obrázek 2.4: Schéma zapojení MCU	14
Obrázek 2.5: Hotový prototyp časovače.....	15
Obrázek 3.1: Vývojový diagram hlavní smyčky programu (pro jeden kanál)	16
Obrázek 3.2: Debouncing softwarovým posuvným registrem	19
Obrázek 4.1: Blokové schéma programovacího modulu.....	20
Obrázek 4.2: Blokové schéma zapojení I2C v programovacím modulu	21
Obrázek 4.3: Schéma zapojení procesorové jednotky	22
Obrázek 4.4: Struktura vnitřní paměti programovacího modulu	23
Obrázek 4.5: OLED displej	23
Obrázek 4.6: Schéma zapojení kapacitního tlačítka	24
Obrázek 4.7: Význam jednotlivých tlačítek	24
Obrázek 4.8: Bluetooth modul HC-05	25
Obrázek 4.9: Převodník úrovní 5 V/ 3,3 V	25
Obrázek 4.10: Schéma zapojení UART/USB převodníku.....	26
Obrázek 4.11: Napájecí modul	27
Obrázek 4.12: Schéma měniče napětí.....	27
Obrázek 4.13: Srovnání PWM a PFM režimů	28
Obrázek 4.14: Finální prototyp programovacího modulu	29
Obrázek 5.1: Mapa ovládacího rozhraní programovacího modulu	34

Obrázek 6.1: Aplikace Terminal pro MS Windows	37
Obrázek 7.1: Aplikace Bluetooth Control app pro Android	39

SEZNAM TABULEK

Tabulka 2.1: Specifikace zadání RC časovače	10
Tabulka 2.2: Základní parametry mikrokontroleru ATmega8A-MU	13
Tabulka 3.1: Přehled využitých periférií mikrokontroleru	17
Tabulka 4.1: Základní vlastnosti mikrokontroleru ATmega644P	21
Tabulka 5.1: Přehled využitých periférií mikrokontroleru	30
Tabulka 5.2: Přehled komunikačních příkazů pro ovládání programovacího modulu...	36

ÚVOD

Časovače pro RC modely jsou elektronická zařízení sloužící k řízení rádiem neřízených modelů. Časovače mají ve své vnitřní paměti uloženou sekvenci časových úseků, podle kterých elektronika časovače natáčí hřídele serv a nastavuje regulátor otáček.

Tato zařízení byla především využívána v minulosti v mechanické podobě pro řízení letadel díky finanční nedostupnosti radiových komponentů do modelů. V současné době se RC časovače využívají spíše pro speciální aplikace nebo pro závody neřízených větroňů.

V první části se práce zabývá vysvětlením problematiky řízení serv a regulátoru v RC modelu. Následně je diskutován a realizován kompletní návrh časovače pro RC modely. Srdcem prototypu je moderní 8 - mi bitový mikrokontroler ATmega8 a EEPROM paměť 24LC64. Mezi hlavní přednosti této součástkové základny je velká paměť pro uložení časových záznamů, nízká spotřeba a nízká hmotnost. Časovač je navržen tak, aby bylo možné využít jak napětí z BEC regulátoru 5 V, tak i napětí z jedné Lion baterie 3,7 V. Komunikace mikroprocesoru a paměti je řešeno pomocí sběrnice I2C. Díky použití této sběrnice je možné pohodlně programovat paměť novými časovými záznamy.

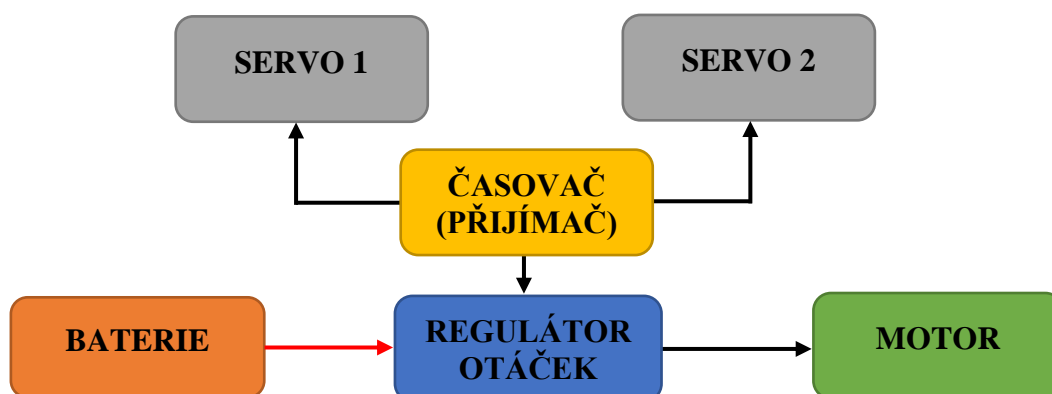
Ve třetí části se práce věnuje ovládacímu firmwaru mikrokontroleru. Návrh programu pro mikrokontroler je prováděn v „Atmel Studio“ od firmy Microchip. Program zahrnuje funkce, jako jsou např. generování PWM signál pro výstupy, čtení a kontrola EEPROM paměti a signalizace stavů na stavové led diodě.

Dále se práce zabývá návrhem a realizací programovacího modulu pro programování časovače. Srdcem zařízení je mikrokontroler ATmega644P vybavená dvěma UART rozhraními. Modul je vybaven komunikačním Bluetooth module a USB, určené pro nahrávání časových sekvencí. Zařízení dále obsahuje vlastní 512 kbit EEPROM paměť, která slouží pro uchování předchozích naprogramovaných časových sekvencí. Díky takovému řešení je možné do RC časovače naprogramovat předchozí časové sekvence bez použití telefonu nebo počítače. Ovládání je zajišťují čtyři kapacitní tlačítka a grafického OLED displeje. Modul je napájen 5 V pomocí měniče napájeného jedním Li-on článkem 18650. Spotřeba zařízení je 60 mA a na jedno nabití schopno fungovat více než 20 hodin.

V závěru práce je stručné shrnutí, jakých výsledků bylo v této práci dosaženo.

1 ČASOVAČ PRO RC MODELÝ

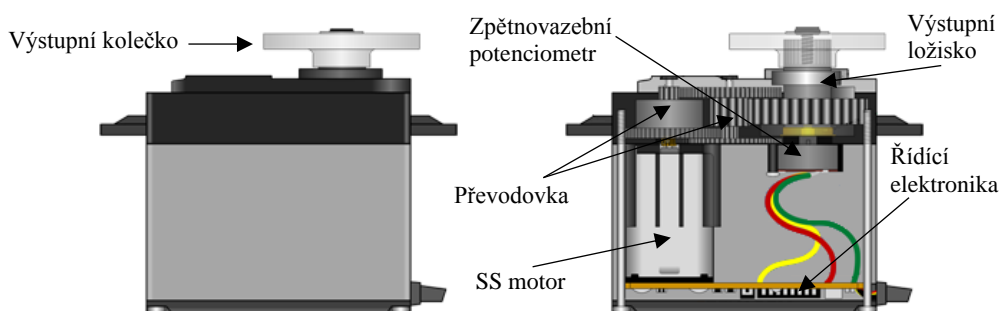
Na základě zadání a obecně známých informací o koncepci vnitřního zapojení RC modelů bylo vytvořeno na obr. 1.1 blokové schéma, které obsahuje všechny základní funkční prvky každého dálkově řízeného modelu letadla. V následujících podkapitolách jsou popsány jednotlivé části podrobně.



Obrázek 1.1: Blokové schéma RC modelu

1.1 Servomotory

Nedílnou součástí každého dálkově řízeného modelu letadla (široké využití mají i v robotice) jsou servomotory (dále jen serva), která zajišťují během letu natáčení klapky na křídlech, směrového kormidla a výškového kormidla. Servo v závislosti na řídicím signálu, které dostává z přijímače, natáčí výstupní hřídel o požadovaný úhel. Citlivost natočení serva je dána kvalitou řídicí elektroniky (viz. kap 1.2). Standardně jsou serva napájena napětím z regulátoru BEC, pracovní napájení serv je 4,8 až 6 V, typicky 5 V. Spotřeba závisí na typu (resp. maximální tažné síly) použitého serva. Po konstrukční stránce se skládá ze tří hlavních částí: elektroniky, převodovky a stejnosměrného motoru (viz. Obr 1.2).



Obrázek 1.2: Řez modelářským servem

Po elektrické stránce je možné serva rozdělit do dvou skupin s digitálním a analogovým řízením (viz kap. 1.2). Mechanické uspořádání digitálních i analogových serv je totožné. Liší se zejména v řídicí elektronice. Deska analogového řízení může být tvořena několika hradlovými obvody, zatímco řídicí desku digitálního serva tvoří mikroprocesor. Servo je přesný zpětnovazební systém. Výstupní hřídel je pevně spojena se zpětnovazebním potenciometrem a podle poměru odporu jeho dráhy lze dosáhnout přesného natočení. Díky tomu, že řídicí elektronika vyhodnocuje právě poměr jezdce na odporové dráze, je eliminována teplotní závislost potenciometru a je možné serva využívat v teplotním rozsahu cca -20 až 60 °C. U vyšších teplot a zvláště při velkém zatížení je riziko tepelného poškození motoru, naopak při teplotách pod -20 °C hrozí zatuhnutí vazelíny v převodovce.

Mezi základní parametry při výběru serva patří krouticí moment, váha, velikost a rychlost otáčení výstupní hřídele. Výběr je vždy otázkou kompromisu. Například pokud požadujeme servo s velkým krouticím momentem a s malým rozměrem, tak bude „pomalé“, protože převodovka bude konstruována do síly.

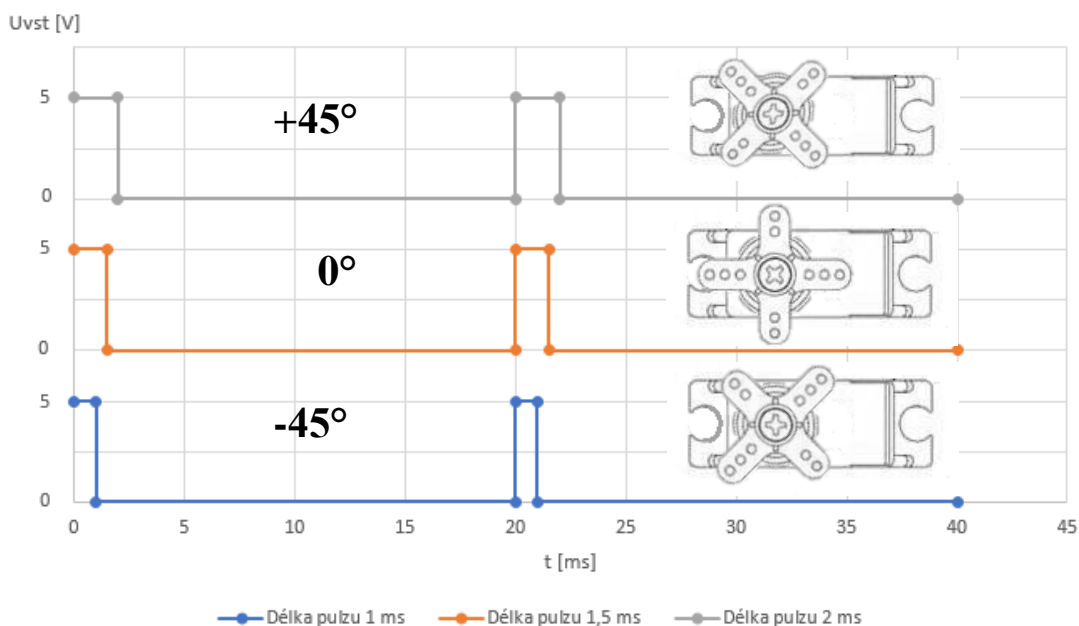
1.2 Řízení serv a regulátoru motoru

Vstupním signálem, na který jednotlivá serva a regulátor reaguje, je výstupní signál z modelářského přijímače. Serva i regulátory mohou mít dva typy řízení: starší a jednodušší analogové nebo novější digitální.

1.2.1 Analogové řízení

Jedná se o pulzní šířkovou modulaci (PWM) s amplitudou 5 V, nosným kmitočtem 50 Hz a šířkou impulzů 1 - 2 ms, která závisí na aktuální poloze páček na modelářské vysílačce. Šířka pulzů odpovídá u serv úhlu otočení 90° a u regulátoru možnosti regulace výkonu 0 – 100 %. Čas 1 ms odpovídá u serv úhlu otočení výstupní hřídele na -45°, středová poloha odpovídá šířce pulzu 1,5 ms a 2 ms odpovídají úhlu otočení na +45°. Situaci nastiňuje obr. 1.3. Serva různých značek mají i nepatrně různé časování. Například při srovnání dvou běžných modelářských serv firmy Hitec HS-81 a HS-422, délka šířky pulzu 1,9 ms odpovídá u prvního serva úhlu natočení +45°, zatímco u druhého serva odpovídá úhlu natočení +40°. Citlivost serv a regulátorů závisí především na ceně, obecně platí, že čím více je zařízení citlivé, tím jemněji se dá úhel natočení (nebo výkon) regulovat. U obyčejných levných serv se práh necitlivosti pohybuje okolo 8-10 µs (odpovídá to

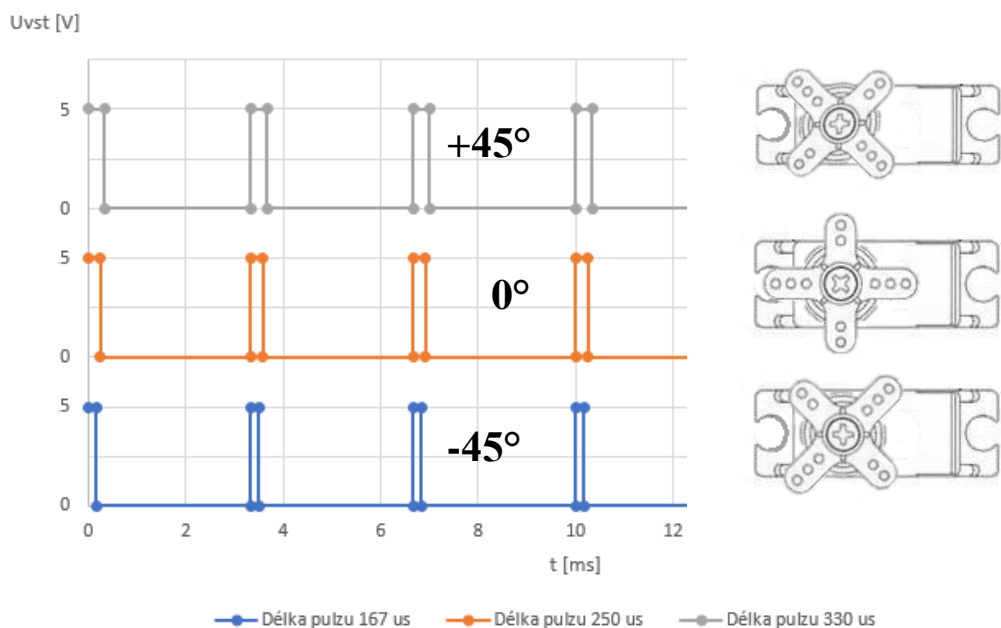
natočení asi $0,9^\circ$) a u dražších bývá i menší než $4\ \mu\text{s}$ (odpovídá to natočení asi $0,4^\circ$).



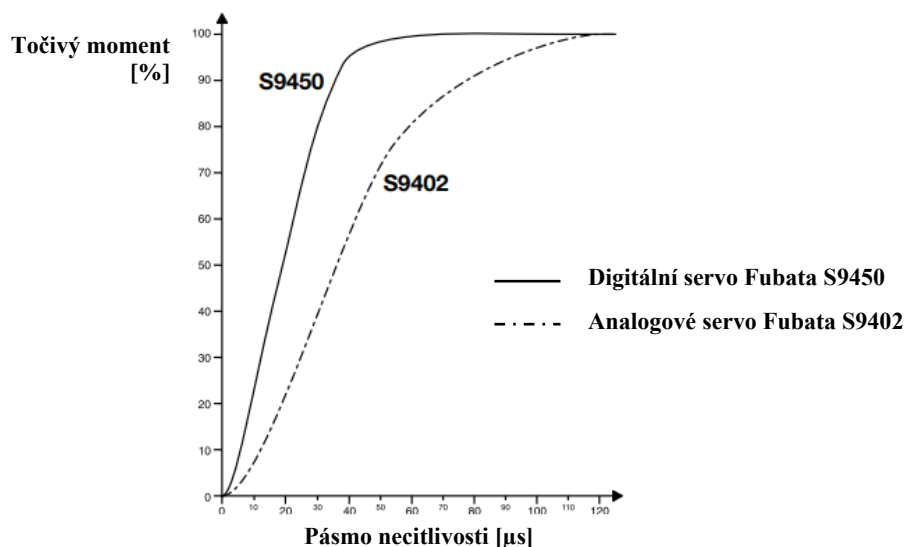
Obrázek 1.3: Řídící impulzy analogového serva

1.2.2 Digitální řízení

Podobně jako u analogového řízení (viz kap. 1.2.1) jsou i digitálně řízená serva a přijímače řízeny PWM modulací s amplitudou 5 V. Hlavním rozdílem je časování vstupního signálu. Nosný kmitočet je 300 Hz a délka řídicích pulz je 167 – 333 μs . Toto řešení má v porovnání s analogovým řízením hned několik výhod, tou hlavní je vyšší rozlišení (servo má menší práh necitlivosti), dále menší délka odezvy na změnu vstupního signálu, vyšší točivý moment, „tužší“ statická poloha a zpětná kompatibilita s analogovým řízením. Některá digitální serva jsou programovatelná. Lze u nich naprogramovat různá nastavení jako například neutrální polohu, koncové body, rychlost pohybu a smysl otáčení. Nevýhodou je vyšší spotřeba.



Obrázek 1.4: Řídící impulzy analogového serva

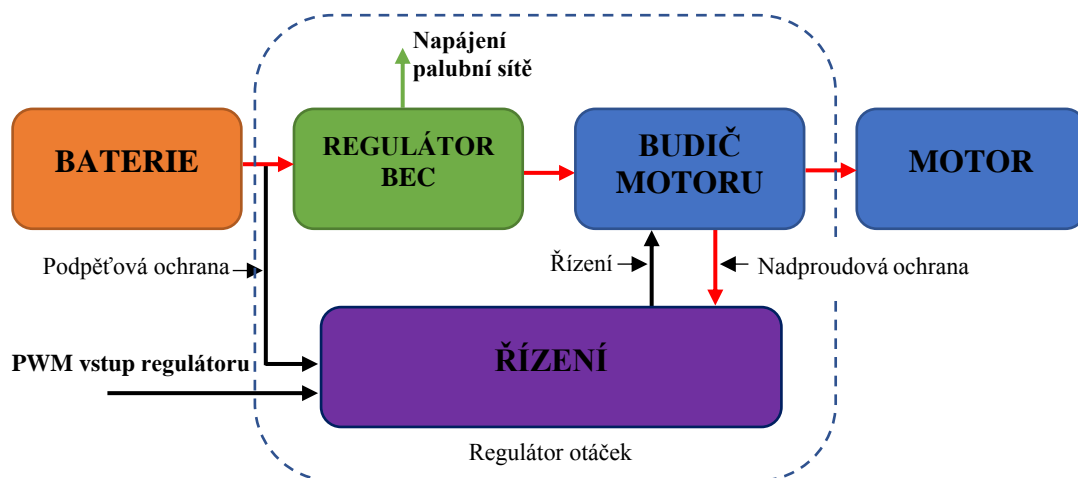


Obrázek 1.5: Srovnání pásma necitlivosti analogového a digitálního serva (převzato a přeloženo z [1])

1.3 Regulátor otáček

Regulátory otáček jsou zařízení, která se používají k řízení otáček modelářských motorů. Podle druhu motoru se dělí na dvě základní skupiny. Jedná se o regulátory pro stejnosměrné a střídavé motory. Podle použití se navíc tyto skupiny ještě dělí na jednosměrné a obousměrné. Jednosměrné regulátory se využívají k řízení motoru jedním směrem, tedy tam kde není třeba zpětného chodu jako například u modelů letadel.

Obousměrné regulátory umožňují i obrácený chod a brzdění, jsou proto vhodné pro použití zejména u modelů aut a lodí.



Obrázek 1.6: Blokové schéma regulátoru motoru

Kromě funkce regulace motoru navíc většina dnes dostupných regulátorů obsahuje obvod BEC. Tento obvod zajišťuje stabilizované napětí pro napájení palubní sítě modelu (serva, přijímač atd.). Využití regulátoru vybaveného tímto obvodem je zvláště výhodné u modelů letadel, kde díky tomuto řešení není nutné použít žádný další zdroj pro napájení (např. přijímače) a tím se ušetří hmotnost modelu. Obvod BEC je tvořen většinou 5 V stabilizátorem napětí. Důležité je, aby tento obvod pokryl spotřebu všech palubních spotřebičů a odstranil rušení způsobené motorem a jeho buzením, které by mohlo mít za následek nekorektní funkci serv (nejčastěji se tento problém projevuje neustálým schvéním výstupní hřídele) a přijímače.

Některé regulátory obsahují obvod PCO, který zajišťuje funkci podpěťové ochrany. Tato funkce se hodí zejména u modelů napájených z Lion nebo Lipol článků. Protože pokud jejich napětí klesne pod minimální úroveň, dochází k trvalému a nevratnému poškození článků. Pokud při provozu RC modelu se začne napětí na baterii přibližovat minimálnímu, začne PCO obvod omezovat výkon regulátoru, tak aby baterie vydržela co nejdéle. Při dosažení kritické hranice napětí je odpojen motor, ale palubní napájení je stále aktivní, tak aby nedošlo k neřízenému pádu modelu letadla.

Podobně jako u serv s digitálním řízením je možné některé typy přijímačů programovat, jedná se o nastavení typu baterie, kontrolu otáček (funkce pro vrtulníky), rozběh motoru, vypínací napětí atd. Lze to pomocí plynové páky vysílače a pomocí zvukových signálů (vysokých, nízkých, krátkých a dlouhých tónů, které vydává připojený motor). V návodu k regulátoru potom bývá tabulka, jak kterou funkci nastavit, což se provádí přesunem plynové páky. Je to velice zdlouhavé a nepřehledné, daleko lepší je nastavení pomocí programovací karty. Ke každému typu regulátoru bývá vlastní karta, buď s propojkami, nebo s displejem. Pro hobby použití stačí základní typ, protože regulátor většinou nastavujeme pouze jednou.

Při výběru regulátoru je nutné brát ohled zejména na jeho proudovou zatížitelnost, typ motoru, pro který je určen a jeho napájecí napětí. Regulátory bývají standardně vybaveny tepelnou i proudovou pojistkou. Tyto pojistky chrání regulátor proti přetížení a jeho následnému zničení.

1.4 Přijímač pro RC modely

Přijímač je srdce každého rádiem řízeného modelu. Jedná se o obvod, který zpracovává vstupní vysokofrekvenční signál z antény a na jeho základě generuje řídicí signály pro jednotlivá serva a regulátor otáček (viz. kap 1.2).

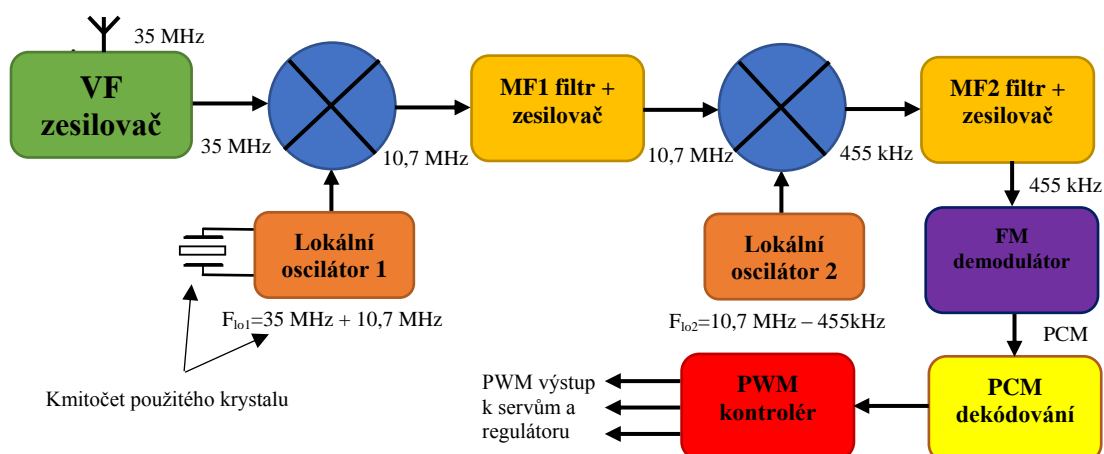


Obrázek 1.7: RC přijímač od firmy Hitec (převzato z [2])

V minulosti přijímače pracovaly na principu superheterodynu s jedním nebo dvojím směřováním řízeného krystalem v pásmu krátkých vln [3]. Pro tyto přijímače byly vyhrazeny kmitočty 35 – 35,91 MHz (pro modely letadel) a 40 – 40,985 MHz (pro modely aut a lodí). Dělení na dvě kmitočtová pásma umožňuje současný provoz všech tří druhů modelů současně. Každé pásmo je dále děleno na více jak 20 kanálů s odstupem 10 kHz. Vyloučení rušení modelů pracujících současně na stejném pásmu je zajištěno pomocí krystalu (shodného ve vysílači i přijímači), který je vázán na konkrétní kanál.

Úplně první a později ty nejobyčejnější přijímače pracovaly s amplitudovou modulací. Použití této modulace není pro RC modely příliš vhodné, protože je velmi náchylná na rušení (zejména na elektrostatické rušení). V současné době jsou dostupné dvě kategorie přijímačů.

První kategorii dnes už méně využívanou tvoří krátkovlnné zařízení s kombinací frekvenční a pulzně kódové modulace [4]. Princip funkce přijímače znázorňuje obrázek 1.8. V závislosti na cenové kategorii přijímače je přijímač realizován jako superheterodyn buď s jedním směřováním nebo s dvojím směřováním (mají nižší šumové číslo). Signál z antény prochází přes vstupní VF zesilovač do směšovače. Signál je směšován společně se signálem harmonického lokálního oscilátoru řízeného krystalem na první mezifrekvenční kmitočet (typicky 10,7 MHz). Kmitočet krystalu udává, na jakém kanálu přijímač pracuje. Signál dále prochází přes mezifrekvenční filtr, který vybere požadovaný signál ze směsky signálů vzniklých ve směšovači.



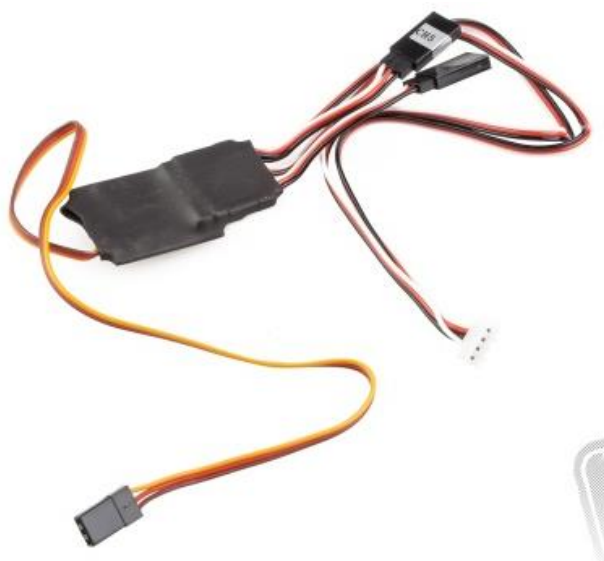
Obrázek 1.8: Blokové schéma FM/PCM přijímače s dvojitým směřováním

Zpravidla se používá spodní kmitočet, k jeho selekci je nejvhodnější dolní propust. Následuje druhý mezifrekvenční stupeň, který signál smísí se signálem generovaného druhým lokálním oscilátorem na kmitočet 455 kHz. Dalším blokem řetězce je FM demodulátor. Existují dvě principiálně odlišné metody demodulace kmitočtově modulovaného signálu: tzv. diskriminátory, které využívají sklonu boku rezonanční křivky oscilačního obvodu k převodu frekvenční modulace na amplitudovou a tzv. fázový závěs. V modelových přijímačích je demodulace řešena fázovým závěsem, protože ke své funkci nepotřebuje indukčnosti a je dostupný v integrované podobě. Na výstupu FM demodulátoru je již obdélníkový pulzně kódovaný signál, ten je PCM dekodérem zpracován a jeho výstupem jsou konkrétní 8 bitové polohy natočení jednotlivých serv. Posledním blokem je PWM kontrolér, který kóduje vstupní hodnoty do PWM signálu.

Druhou kategorií jsou mikrovlnné přijímače, které pracují v ISM pásmu 2,4 GHz. V současné době jsou nejrozšířenější mezi modeláři, protože mají hned několik výhod při porovnání s krátkovlnnými přijímači. Tou hlavní je prakticky žádná šance rušení dvou modelů. Přijímače nepoužívají krystal pro volbu kanálu, ale mají obousměrnou komunikaci s vysílačem. Při zapnutí vysílače prohledá volné kanály a na základě toho se obě strany dohodnou na kanálu, na kterém budou vysílat. Pokud se během vysílání objeví rušení, tak se vysílač i přijímač dohodnou na jiném kanálu, změna je možná i několikrát za vteřinu. Další nespornou výhodou je, že použití mikrovln umožňuje použít krátké antény. Délky antény $\lambda/4$ pro pásmo 35 MHz vychází 2,14 m, zatímco pro pásmo 2,4 GHz jen 0,31 m.

1.5 Časovač pro RC modely

Časovač pro RC modely je elektronické zařízení určené pro rádiem neřízené modely rádiem. Jedná se o zařízení, které je svojí funkcí podobné přijímači (viz. kap. 1.4) s rozdílem, že povely pro serva a regulátor neřídí člověk vysílačem, ale čtou se z předem naprogramované paměti umístěné přímo v modelu. Elektronické časovače se používají například při závodech rádiem neřízených modelů letadel nebo obecně, kde není možné použít rádiem řízený model (například z důvodu širokopásmového rušení). V minulosti, když neexistovaly elektronické časovače, se používaly mechanické časovače, tvořené buď mechanickým hodinovým strojkem nebo doutnákem, který když uhořel, tak např. natočil výškové kormidlo letadla pro ukončení letu. Toto časování bylo nepřesné, neumožňovalo více časových úseků.



Obrázek 1.9: Komerčně prodávaný časovač Valkyrie pro RC modely (převzato z [5])

Časovač je napájen z obvodu BEC. Spouštění startu časovače se liší podle druhu letadla. U motorových modelů (házených z ruky) je možné časovač spustit těsně před letem pomocí tlačítka umístěného přímo na modelu letadla. Například u větroňů bez motoru je možné spustit časovač bezdrátově pomocí infračerveného ovladače.

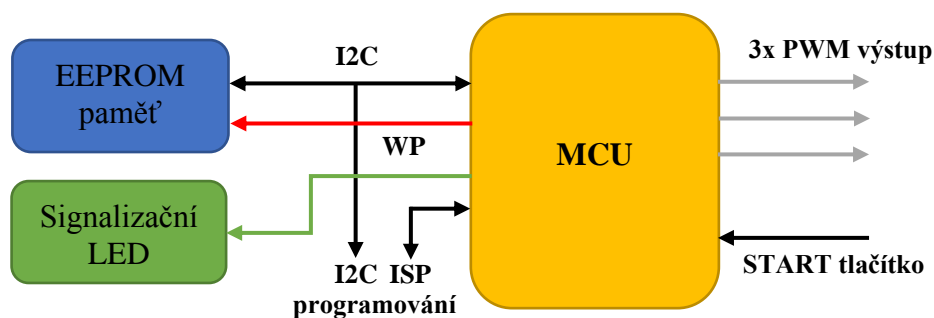
Na první pohled je zřejmé, že využití časovače v modelech letadel na složitější manévry během letu je zcela nevyzpytatelné a nevhodné. Časovač není autopilot a pokud například během letu zafouká vítr, dojde k nekorigované odchylce od ideálního kurzu s možností pádu.

Při výběru časovače patří mezi nejdůležitější parametry pro kolik serv (regulátorů) je zařízení určeno, kolik se vejde časových úseků pro každý PWM kanál do paměti, váha, proudová spotřeba a způsob programování.

Programování komerčně dostupných časovačů je poměrně složité pomocí tlačítka a ledky, podobně jako programování regulátor (viz kap. 1.3).

2 NÁVRH HARDWARE

Na základě prostudování dostupných informací o RC časovačích bylo vytvořeno následující schéma na obrázku 2.1. Zařízení se skládá ze dvou hlavních částí, z mikroprocesoru a paměti EEPROM.



Obrázek 2.1: Blokové schéma RC časovače

2.1 Požadavky na návrh časovače

Na základě odborných konzultací byly specifikovány přesné požadavky na výsledný prototyp (viz. tab 2.1).

Tabulka 2.1: Specifikace zadání RC časovače

Druh rozhraní řízených serv/regulátoru	Analogový
Typ procesoru	8-bitový AVR
Velikost napájecího napětí [V]	3,2 – 5,5 V
Nejmenší časové rozlišení (nejkratší doba kterou může trvat jeden záznam) [s]	0,1
Počet PWM výstupů	1x regulátor otáček + 2x servo
Počet záznamů pro PWM výstup pro regulátor otáček	20
Počet záznamů pro každé servo	10

Dále byly vytvořeny dva testovací scénáře umístěné v přílohách: A1 – letový scénář INDOOR a A2 – letový scénář OUTDOOR.

2.2 Výpočet a volba vhodné EEPROM paměti

Dle požadavků v tabulce 2.1 je patrné, že je potřeba přepisovatelná paměť pro minimálně čtyřicet záznamů s minimální délkou trvání záznamu 100 ms (nejkratší doba, kterou může trvat jeden záznam). Rovnice 2.1 zahrnuje výpočet maximální délky trvání jednoho záznamu za předpokladu, že budou použita dvou bajtová čísla. Z kapitoly 1.2.1 vyplývá, že maximální délka střidy je 2 ms s nejmenším krokem 10 us. Rovnice 2.2 řeší velikost použité proměnné.

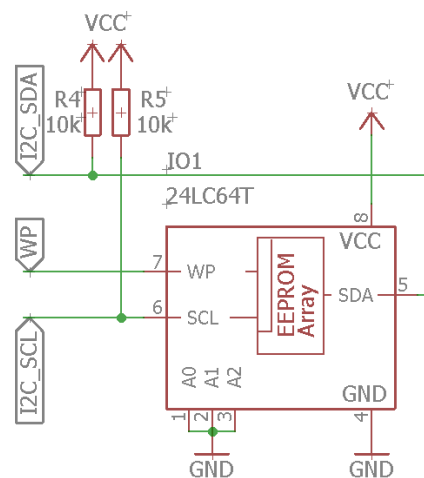
$$t = (2^{bit} - 1) \cdot \Delta t_{min} = (2^{16} - 1) \cdot 0,1 = 6563,5 \text{ s} \quad (2.1)$$

$$\text{počet bitů} = \sqrt{\frac{t_{max}}{\Delta t_{min}}} = \sqrt{\frac{2 \cdot 10^{-3}}{10 \cdot 10^{-6}}} = 14,14 \doteq 16 \quad (2.2)$$

Z výsledků rovnic vyplývá, že je nutné čísla v paměti organizovat jako dvou bajtová, tak aby zařízení splňovalo požadované vlastnosti. Výpočet celkové velikosti minimální potřebné paměti zachycuje následující rovnice:

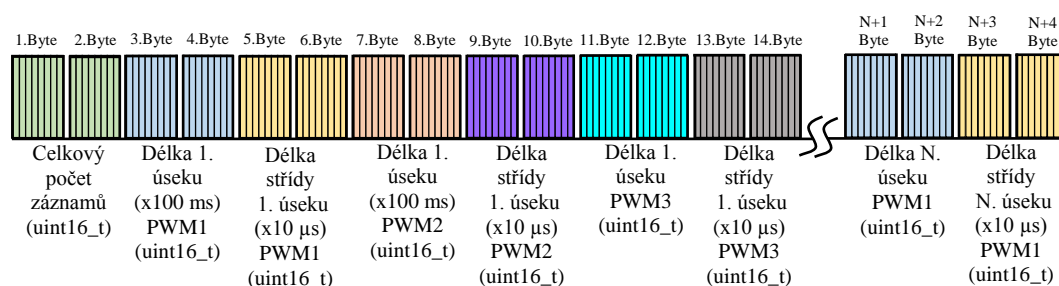
$$\text{velikost paměti} = \text{počet záznamů} \cdot \text{velikost čísel} \cdot 2 = 40 \cdot 2 \cdot 2 = 160 \text{ B} \quad (2.3)$$

Celková velikost paměti vypočtená podle rovnice 2.3 není vůbec kriticky velká a vejde se do vnitřní paměti EEPROM všech dostupných AVR na trhu. S ohledem na větší univerzálnost časovače (uložení více záznam nebo pozdější implementace více kanálů) byla zvolena externí sériová EEPROM paměť 24LC64 [6]. Jedná se o polovodičovou paměť typu EEPROM, tudíž v ní zůstávají data zachována i po odpojení napájecího napětí. Tato paměť má velikost 64kbit (8 kB) komunikuje přes sběrnici I2C a pracuje s napájecím napětím 2,5 a 5,5 V. Je od výrobce Microchip. Velkou výhodou této paměti je velmi nízká spotřeba, typický odběr je 50 nA při čtení a 100 μA při zápisu. Další výhodou je pin Write-Protect, díky němu může mikrokontroler zablokovat neautorizovaný přepis paměti. Integrovaný obvod je dostupný v pouzdru DFN 8, které díky svým rozměrům 2x3 mm je ideální pro tuto aplikaci.



Obrázek 2.2: Schéma zapojení paměti EEPROM

Na obrázku 2.2 je schéma zapojení paměti. Vstup SCL (hodiny) jsou připojeny ke vstupu SCL mikroprocesoru a obdobě datový vstup SDA paměti ke vstupu SDA na procesoru. Rezistory R4 a R5 jsou pull-up rezistory, které jsou nutné pro korektní funkci sběrnice I2C. Adresové piny A0 - A2 se používají k odlišení zařízení připojených k I2C sběrnici. Protože v časovači je použita pouze jedna I2C periferie, jsou všechny adresové piny připojeny ke GND (z důvodu co nejjednoduššího návrhu DPS), podle datasheetu [4] toto zapojení odpovídá I2C adrese 0xA0. Na obrázku 2.3 je vnitřní organizace paměti záznamů časovače.



Obrázek 2.3: Vnitřní organizace paměti záznamů časovače

Výpočet maximální počtu paměťových pozic pro každý kanál je proveden v rovnici 2.4. Tento výpočet je pouze rámcový, ve skutečnosti může být u některého z kanálů vyšší, protože v praxi většinou má každý kanál různý počet uložených časů (viz. tab. 2.1).

$$\text{Počet záznamů na kanál} = \frac{\frac{\text{Kapacita EEPROM} - 2}{\text{Počet kanálů}}}{\text{Počet bajtů na kanál}} = \frac{\frac{8000 - 2}{3}}{4} \doteq 666 \quad (2.4)$$

2.3 Mikrokontroler ATmega8

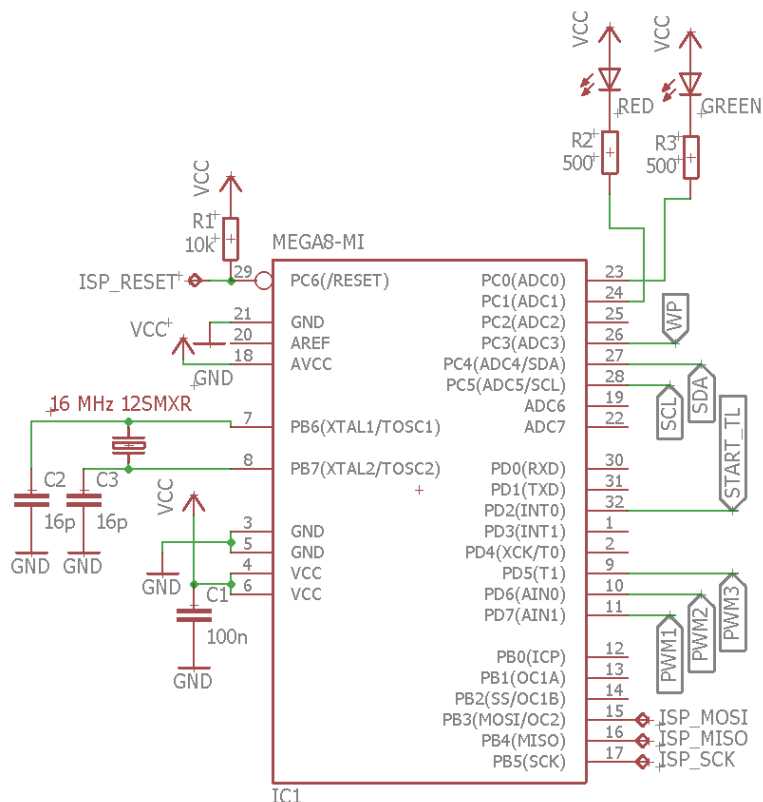
Základním řídicím prvkem celé aplikace je jednočipový mikroprocesor. V současné době je na trhu dostupná celá řada nejrůznějších typů od různých výrobců. Mezi nejvýznamnější výrobce patří například Microchip, NXP, Intel a STM. Vzájemně se od sebe liší svými parametry, vybavením a v neposlední řadě také cenou. V současné době jsou velmi rozšířeny procesory ARM, které disponují lepšími periferiemi i vyšší výpočetním výkonem za srovnatelnou cenu jako starší procesory AVR. Pro tuto aplikaci byl vybrán na základě zadání procesor AVR, dle těchto požadavků:

- nízká spotřeba
- napájecí napětí 3,2 - 5,5 V
- miniaturní provedení (nízká hmotnost)
- sběrnice I2C pro připojení externí paměti
- programovací rozhraní ISP/JTAG
- interní časovače (minimálně 2)
- velikost flash paměti 8 kB

Vzhledem k rozšířenosti a výborné dokumentaci od výrobce byl vybrán procesor ATmega8A-MU [7] z rodiny procesoru AVR od společnosti Microchip (dříve Atmel). Tento procesor disponuje dostatečnou programovou pamětí, velkým množstvím I/O pinů a umožňuje připojení externí EEPROM paměti přes I2C sběrnici. Z důvodu co nejmenších rozměrů a váhy finálního prototypu bylo zvoleno pouzdro VQFN32 s rozměry 5 x 5 mm. Základní parametry mikrokontroleru ATmega8A-MU jsou uvedeny v následující tabulce 2.2.

Tabulka 2.2: Základní parametry mikrokontroleru ATmega8A-MU

Parametr	Hodnota
Typ programové paměti	Flash
Velikost programové paměti [kB]	8
Výpočetní výkon CPU [MIPS]	16
Velikost RAM [B]	1024
Velikost EEPROM [B]	512
Komunikační rozhraní	1x UART, 1x SPI (ISP), 1x I2C
Celkový počet pinů	32
Časovače	2x 8-bitový, 1x 16-bitový
Napájecí napětí [V]	2,7 – 5,5
Spotřeba [mA]	10
Typ pouzdra	VQFN32
Pracovní rozsah teplot [°C]	-45 až 85



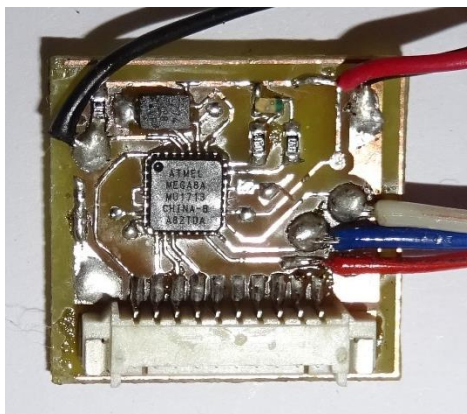
Obrázek 2.4: Schéma zapojení MCU

Na obrázku 2.4 je schéma zapojení mikroprocesoru. Procesor je taktován přesným 16 MHz krystalem 12SMXR. Tento krystal má přesnost ± 30 ppm, což je pro tuto aplikaci plně dostačující. Provedení krystalu bylo především z důvodu požadavku na malé rozměry a váhu zvoleno v SMD pouzdru s rozměry 7,5 x 5,1 x 1,4 mm. Veškeré pasivní součástky jsou v pouzdru 0603. Indikační dvojité led diody slouží pro uživatele ke snadnému zjištění stavu, ve kterém se časovač nachází. Stavy diod jsou detailně popsány v kap. 3. Programovací piny ISP rozhraní jsou realizovány pomocí testovacích plošek přímo na desce plošného spoje.

2.4 Praktická realizace prototypu RC časovače

Časovač je postaven na oboustranné desce plošných spojů o rozměrech 21,3 x 19,6 mm. Výsledná deska včetně osazovacích výkresů je umístěna v příloze B.

Časovač je z důvodu co nejmenší konstrukce osazen pouze součástkami v pouzdech SMD. Důležitým konstrukčním prvkem je konektor, kterým se deska připojuje k modelu. Tento konektor plní dvě funkce, přenáší elektrické signály a zároveň funguje díky svým zámkům i jako držák desky. Na obrázku 2.5 je finální prototyp časovače pro RC modely. V příloze D jsou detailní technické parametry finálního výrobku.



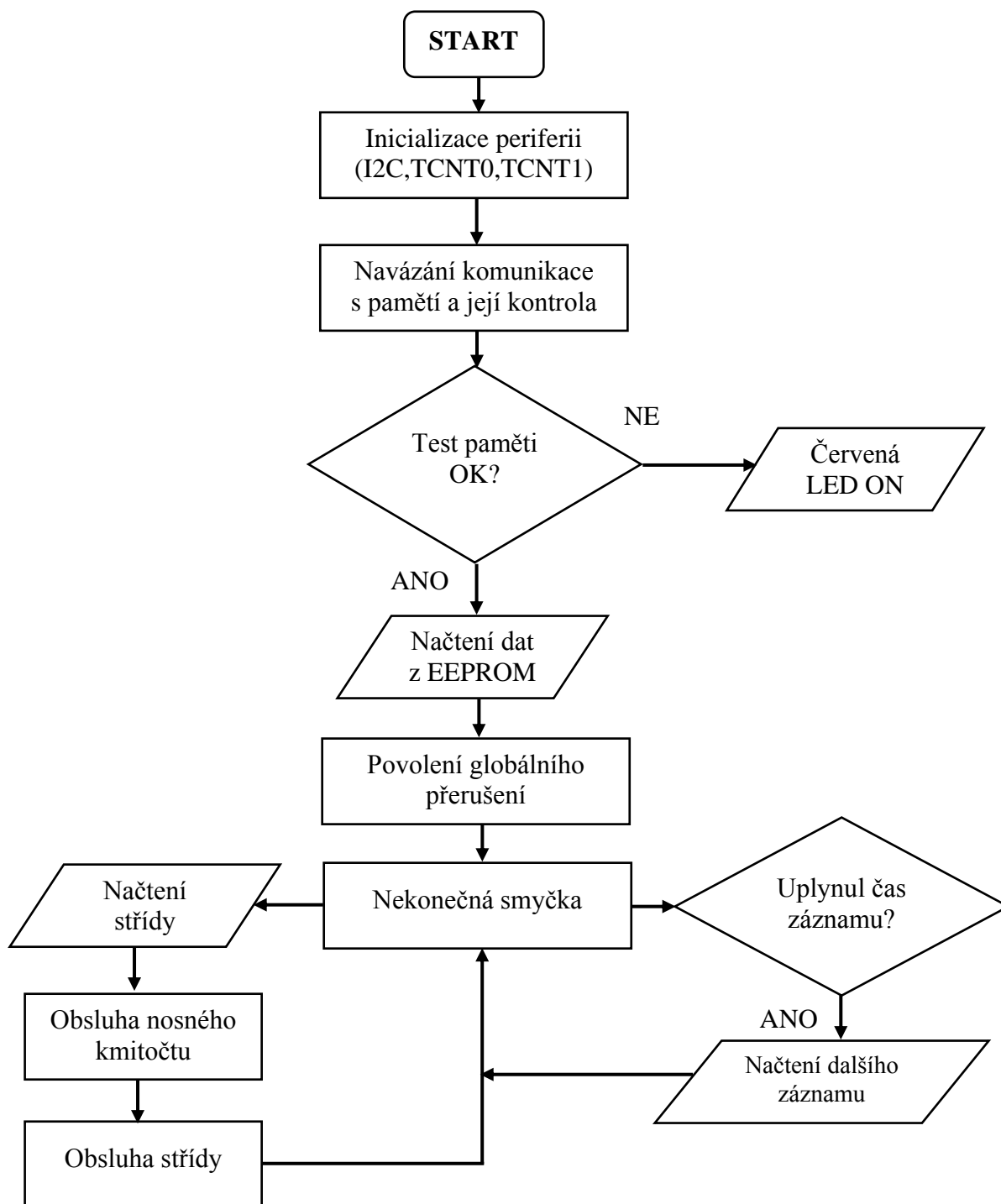
Obrázek 2.5: Hotový prototyp časovače

3 ŘÍDÍCÍ SOFTWARE PRO ČASOVAČ

Firmware pro mikrokontroler časovače byl napsán v jazyce C. Jako vývojové prostředí bylo zvoleno Atmel Studio verze 7.0 [8]. Jedná se o volně šiřitelný nástroj obsahující textový editor se zvýrazněním syntaxe, kompilátor debugger postavený na platformě Visual Studio.

3.1 Běh programu

Na obrázku 3.1 je vývojový diagram hlavní smyčky programu mikrokontroleru pro jeden výstup. Po připojení desky k napájecímu napětí dojde k automatické inicializaci mikrokontroleru, paměti a dojde ke kontrole integrity dat uložených v paměti EEPROM. V případě, že je paměť v pořádku, dojde k nakopírování první sady dat z EEPROM do paměti RAM. Následně program čeká na stisknutí startovacího tlačítka připojeného k pinu 32 (PD2). Při stisku tlačítka start dojde ke spuštění generování pilotního 50 Hz signálu na výstupech PWM1-PWM3 (piny na MCU 11-9) a zároveň se nastaví střída z prvního uloženého záznamu. Po uplynutí času trvání prvního záznamu dojde k načtení dat dalšího záznamu. Po uplynutí všech záznamů je nastavena uživatelsky nastavitelná pozice serv a výkon motoru, která trvá do odpojení časovače od modelu nebo opětovného stisku tlačítka start. Toto řešení umožňuje, aby model letadla mohl bezpečně doplachtit na zem, pokud se náhodou stane, že bude ještě ve vzduchu.



Obrázek 3.1: Vývojový diagram hlavní smyčky programu (pro jeden kanál)

3.2 Funkce programu

Hlavním stavebním prvkem časovače je mikrokontroler ATmega8. Tento mikrokontroler má celou řadu periférií (viz. tab. 2.2). V tabulce 3.1 je výčet využitých periférií v řídicím programu a jejich účel použití.

Tabulka 3.1: Přehled využitých periférií mikrokontroleru

Název periferie	Využití
Timer/Counter0	Hlavní časovač pro modul scheduler.c, (časování nosné, střídý i délek časových úseků)
Timer/Counter1	Časovač pro maskování vstupu tlačítka start
ISP	Sériové programování procesoru
I2C	Sběrnice pro komunikaci s pamětí EEPROM

V následujícím seznamu je uveden výčet všech důležitých funkcí, které časovač využívá. Díky relativní jednoduchosti nebyl program členěn do více modulů. Obsahuje pouze modul **main.c** a knihovny, které obsluhují jednotlivé periferie.

- *main ()* – výchozí funkce programu. Nemá vstupní ani návratovou hodnotu. Po prvotní inicializaci periférií v ní dojde k zacyklení v nekonečné smyčce, ze které jsou volány funkce pro generování PWM modulace servo () a časování hodiny ().
- *servo ()* – stěžejní funkce celého programu. Slouží ke generování pulzně šířkové modulace na jednotlivých výstupních pinech procesoru.
- *hodiny ()* – další neméně důležitá funkce, která slouží k hlídání časových intervalů jednotlivých záznamů a po jejich uplynutí načítá další záznam do paměti.
- *ctieeprom ()* – funkce která načítá z paměti EEPROM dvoubajtová čísla. Vstupní hodnotou je adresa paměti a výstupní je číslo v datovém typu uint16_t.

3.3 Generování výstupního PWM signálu

Z levé části obr. 3.1 vychází funkce *servo()*, která generuje výstupní pulzně modulovaný signál. Při každém zavolání funkce je hodnota proměnné *delay* inkrementována podle proměnné globální proměnné *TickCounter* scheduleru. Každých 20 ms (50 Hz) jsou všechny výstupy a proměnné *ons1-3* nastaveny do log. 1. *Ons1-3* aktivují další neblokující časování, které po uplynutí délky střídý uvede porty opět do log. 0, vynuluje proměnnou *delay* a *ons1-3*. Následující zdrojový kód popisuje pro lepší pochopení problematiky generování signálu pro jeden kanál:

```
void servo(void)
{
    static uint16_t delay = 0; //pomocná proměnná SCHEDULERU
    static volatile bool ons1 = false; //pomocné proměnné pro střídý
    if (HasDelayElapsed(2000, &delay, true)) //obsluha pilotního taktu
```

```

{
    PORTD |= _BV(PD7); //každých 20ms log 1 na výstupech

    onsl=true; //povolení časovačů střídý
}
if (onsl)
if (HasDelayElapsed(EEPROM_lookupS1[(pointerS1*2)-1], &delay,
false)) //obsluha střídý pro S1
{
    PORTD = PORTD & ~_BV(PD7);
    onsl=false;
}
}

```

3.4 Hlídač délky záznamů

Z pravé části obr. 3.1 vychází funkce *hodiny()*, která hlídá délku jednotlivých záznamů a po jejich uplynutí nastaví proměnné *pointerS1-3* na začátek dalšího záznamu. Neblokující zpoždění inkrementuje proměnnou *cas* každých 100 ms. Tzn., že v proměnné *cas* je uložena doba ve stovkách milisekund od stisku tlačítka start. Díky tomu, že proměnná *cas* je 16 bitové nezáporné číslo, je celková délka všech záznamů omezena na délku trvání 109 minut. Proměnná *cti_eeprom* povoluje načtení nového záznamu z EEPROM paměti na předchozí pozici. Následující zdrojový kód popisuje pro lepší pochopení problematiky obsluhu délky záznamů pro jeden kanál:

```

void hodiny()
{
    static uint16_t delay = 0; //pomocná proměnná SCHEDULERU
    static uint16_t cas=0; //čítač času od zapnutí
    static uint16_t casTEMP1=0; //pracovní proměnná
    if (HasDelayElapsed(10000, &delay, true))
    {
        cas++; //inkrementace promenne cas (násobek 100ms)

        if (cas==(casTEMP1+EEPROM_lookupS1[(pointerS1*2)-2]))
        {
            pointerS1++; //posunutí na další pozici v
                        EEPROM_lookupS1
            casTEMP1=cas; //uložení do pracovní proměnné
            adresa[0]=adresa[0]+12; //adresa pro orientaci v EEPROM
            cti_eeprom =true; //povolení načtení dat z EEPROM
        }
        if (pointerS1>5) pointerS1=1;
    }
}

```

3.5 Časovače a přerušení

Z tabulky 3.1 vyplývá, že program pro svůj běh využívá časovače TCN0 a TCN1. Timer 0 je osmibitový časovač s nastaveným přetečením po 10 μ s. Po jeho přetečení nastává přerušení během, kterého je inkrementována proměnná scheduleru TickCounter. Výpočet inicializace je v rovnici 3.1. Časovač využívá děličku osmi, proto $N=8$.

$$f_{OCR0A} = \frac{f_{CLK I/O}}{N \cdot (1 + OCR0A)} \Rightarrow OCR0A = \frac{f_{CLK I/O}}{N \cdot (1 + f_{OCR0A})} - 1 = \frac{16 \cdot 10^6}{8 \cdot \frac{1}{10 \cdot 10^{-6}}} - 1 = 19 \quad (3.1)$$

$$OCR0A = 19 \Rightarrow 0x13$$

Timer 1 je šestnácti bitový časovač. Tento časovač slouží k maskování vstupu pro start tlačítka pomocí posuvného registru pro odstranění zákmitů. Perioda časovače byla zvolena 2,304 ms. Implementace této metody pro tuto aplikaci není nijak kritická na konkrétní periodu časovače, obvykle se volí jednotky ms (viz. kap 3.5). Časovač využívá předděličku šedesáti čtyřmi, proto $N=64$. Vztah jeho výpočtu (viz. rovnice 3.2) je obdobný jako rovnice 3.1.

$$f_{OCR0A} = \frac{f_{CLK I/O}}{N \cdot (1 + OCR0A)} = \frac{16 \cdot 10^6}{64 \cdot (1 + 575)} = 2,304 \text{ ms} \quad (3.2)$$

3.6 Scheduler

Běh celého programu je řízen knihovnou scheduler. Jedná se o neblokující časovač, který je řízen časovačem Timer0. Z principiálního pohledu po inicializaci všech periférií MCU dojde k zacyklení programu v nekonečné smyčce, ze které jsou po sobě volány funkce servo() a hodiny(). Jednotlivé funkce se spouští na základě záznamů uložených v EEPROM paměti. Během toho, co jedna funkce čeká, druhá může být vykonávána. Jednotlivé funkce (úlohy) mají své statické stavové proměnné a sdílí spolu paměť RAM. Jedná se v podstatě o nejjednodušší formu kooperativního RTOS.

3.7 Odstranění zákmitů startovacího tlačítka

Tlačítko start je připojeno k procesoru přes pin PD2 (pin č. 32). V průběhu, kdy je připojené k časovači napájecí napětí může dojít k různým zákmitům způsobených například rušením z regulátoru a časovač by se mohl neplánovaně nebo nekorektně spustit. Proto je nutné podobně jako u různých kritických průmyslových aplikací velmi kvalitního odstranění zákmitů. V této aplikaci je použito digitálního posuvného registru (viz. obr. 3.3).



Obrázek 3.2: Debouncing softwarovým posuvným registrem (převzato z [11])

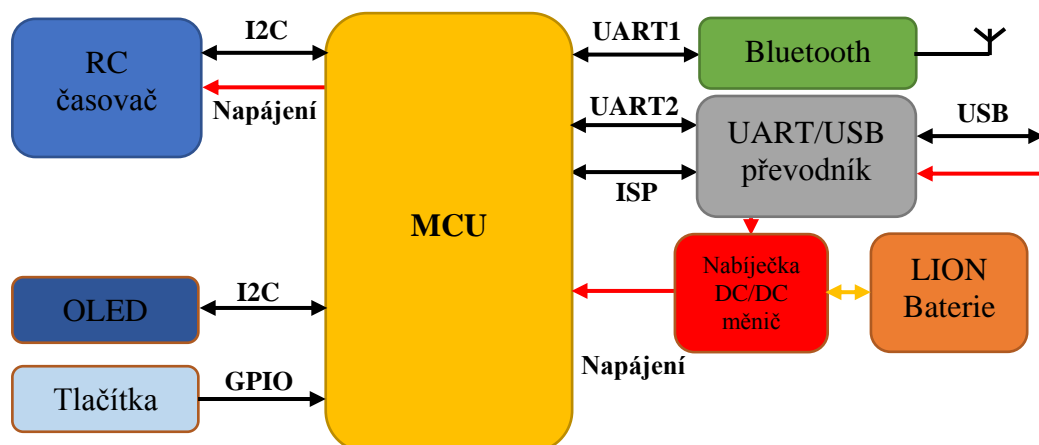
Zdrojový kód pro detekci stisku tlačítka (tj. sestupné hrany logického signálu) v posuvném registru je realizován během přerušení a vypadá následovně:

```
ISR(TIMER1_COMPA_vect)
{
    static uint16_t zakmity = 0xFFFF ;
    zakmity <= 1;
    if (PIND & _BV(PD2)) zakmity |= 0x0001 ;
    if (zakmity == 0x8000) start=true;
}
```

Obsluhu tlačítka zajišťuje Timer1. K přerušení dochází jednou za 2,304 ms. Po stisku tlačítka (viz obr. 3.2) dojde na vstupu MCU k přechodu na log. 0, následnému zakmitání, a nakonec k ustálení na log. 0. Stav tlačítka je postupně vzorkován a vsouván do posuvného registru od nejméně významného bitu. Sestupná hrana je detekována při stavu registru 0x8000, což odpovídá log. 1 (poslední zbytek zákmitů) a následně patnácti log. 0 (ustálený stav stisku). Podle obrázku 3.2 k vyhodnocení sestupné hrany dojde až po posunutí zelené části k nejvýznamnějšímu bitu posuvného registru. V praxi se softwarového odstranění zákmitů využívá vzhledem k ceně součástek více než hardwarového.

4 PROGRAMOVACÍ MODUL

Pro zajištění pohodlného a rychlého programování časovače s ohledem možnost použití v terénu byl navržen modul, do kterého je možné desku časovače zasunout a pohodlně naprogramovat. Na obrázku 4.1 je blokové schéma programovacího modulu.



Obrázek 4.1: Blokové schéma programovacího modulu

Ve vnitřní paměti je možné uložit až 10 sekvencí časových dat včetně jmen. Tyto sekvence je možné libovolně nahrávat do časovače, mazat nebo zpětně exportovat do počítače nebo mobilního telefonu. Na displeji programovacího modulu se dále zobrazuje aktuální čas, stav baterie, detekce připojeného čipu a je umožněno uživateli provádět

základní nastavení. Z hardwarového hlediska se programovací modul skládá z pěti hlavních částí: z procesorové jednotky, Bluetooth rozhraní, OLED displeje, ovládacích tlačítek a obvodů zajišťující napájení zařízení.

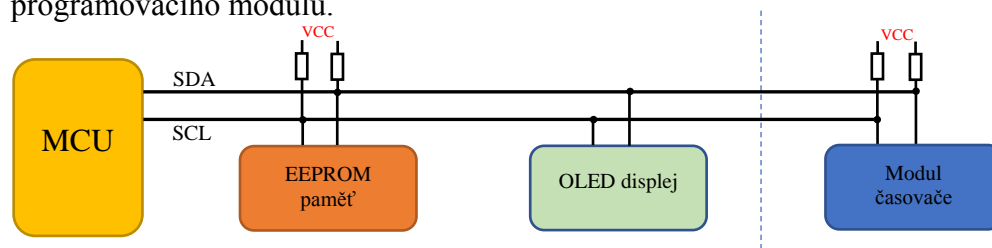
4.1 Procesorová jednotka

Procesorová jednotka je řízena mikrokontrolerem ATmega644P. Tento procesor je taktován krystalem na 16 MHz a obsluhuje všechny funkční prvky programovacího modulu (viz. obr 4.3). Jeho velkou výhodou jsou dvě rozhraní UART, z nichž jedno je využito pro komunikaci s počítačem přes UART/USB převodník a druhé pro komunikaci přes Bluetooth. Při porovnání s mikroprocesorem použitým v časovači, má procesor osmkrát větší paměť, disponuje více periferiemi a větším výpočetním výkonem (viz. tab. 4.1).

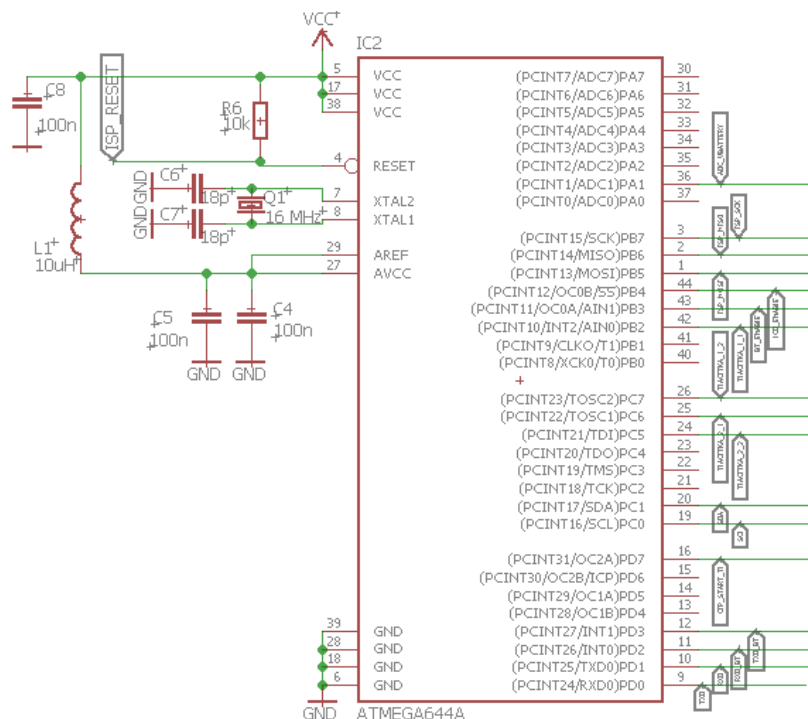
Tabulka 4.1: Základní vlastnosti mikrokontroleru ATmega644P

Parametr	Hodnota
Typ programové paměti	Flash
Velikost programové paměti [kB]	64
Výpočetní výkon CPU [MIPS]	20
Velikost RAM [B]	4096
Velikost EEPROM [B]	2048
Komunikační rozhraní	2x UART, 3x SPI (ISP), 1x I2C
Celkový počet pinů	44
Časovače	2x 8-bitový, 1x 16-bitový
Napájecí napětí [V]	4 – 5,5 (pro $f_{CPU} = 20$ MHz)
Spotřeba [mA]	12
Typ pouzdra	TQFP44
Pracovní rozsah teplot [°C]	-40 až 85

Komunikace mezi MCU, pamětí EEPROM (viz kap. 4.2), displejem (viz kap. 4.3) a modulem časovače je realizována pomocí sběrnice I2C s taktem 100 kHz. Použití této sběrnice má velkou výhodu, protože připojení modulu časovače je typu „plug and play“, tzn. že při připojení není třeba zařízení vypínat, nebo cokoli nastavovat (podobně jako u sběrnice USB). Na obrázku 4.2 je blokové schéma zapojení sběrnice uvnitř programovacího modulu.



Obrázek 4.2: Blokové schéma zapojení I2C v programovacím modulu



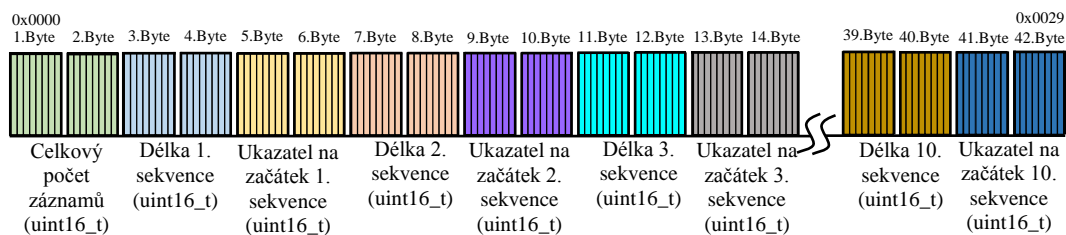
Obrázek 4.3: Schéma zapojení procesorové jednotky

4.2 Struktura vnitřní paměti programovacího modulu

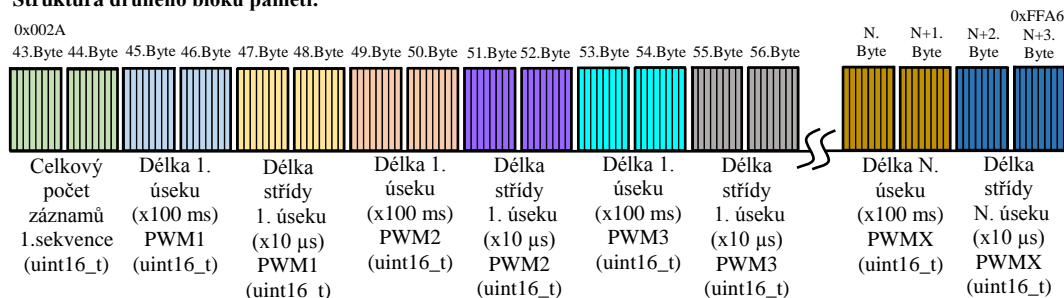
Při výběru typu paměti programovacího modulu byla jedním z hlavních požadavků velikost paměti, aby byla schopná uchovat desetinásobek dat (deset časových sekvencí), který se vejde do paměti modulu časovače. Proto byla vybrána sériová EEPROM paměť 24LC512 od společnosti Microchip. Jedná se o paměť o velikosti 512 kbit (64 kB). Tento typ paměti je zároveň největší dostupnou pamětí, která je 16 bitově adresovaná, protože její kapacita využívá celý adresní prostor. Hardwarově je programovací modul kompatibilní i s většími pamětmi, ale je nutné upravit program mikrokontroleru.

Paměť je rozdělena řídicím programem do tří bloků (viz. obr. 4.4). Všechna data včetně názvů jednotlivých časových sekvencí jsou uložena v 16 bitovém tvaru. V prvních dvou bajtech paměti je hodnota kolik sekvencí paměť obsahuje. Následuje 40 bajtů ukazatelů na jednotlivé časové sekvence. Kde vždy dva bajty udávají délku sekvence a následující dva udávají přímo adresu v paměti, kde sekvence začíná. Další blok (začínající od adresy 0x002A) velký 63,87 kB tvoří oblast dat, ve které jsou uloženy za sebou vlastní časové sekvence ve stejném formátu jako v modulu časovače (viz. obrázek 2.3). Poslední blok (začínající od adresy 0xFFA7) je tvořen 88 bajty jmen jednotlivých sekvencí. Každé jméno tvoří maximálně sedm znaků bez diakritiky

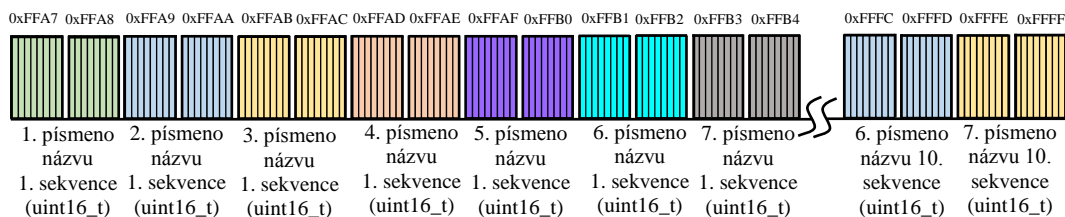
Struktura prvního bloku paměti:



Struktura druhého bloku paměti:



Struktura třetího bloku paměti:



Obrázek 4.4: Struktura vnitřní paměti programovacího modulu

4.3 OLED displej

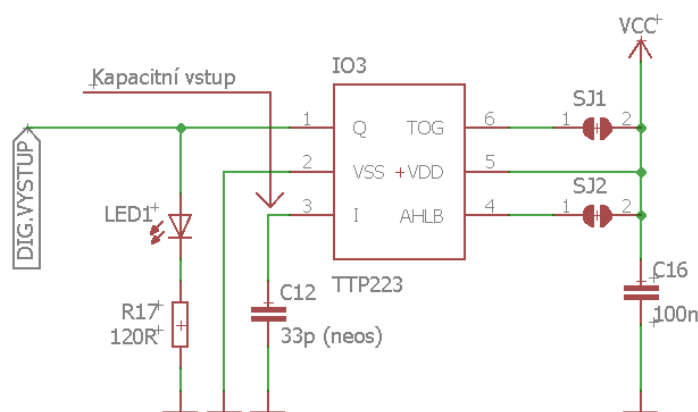
Jako zobrazovací prvek je použit monochromatický OLED displej s úhlopříčkou 0,91 palce (viz. obr 4.5). Rozlišení tohoto displeje je 128 x 32 pixelů a o řízení celého displeje se stará řadič SSD1306, který komunikuje po rozhraní I2C. Tento typ může mít bílé nebo modré podsvícení a pro jeho napájení lze použít napětí v rozsahu 3,3 až 5 V. Velkou výhodou OLED technologie je velmi nízká spotřeba, proudový odběr se zvyšuje s počtem rozsvícených bodů. Maximální odběr se pohybuje okolo 25 mA při napájení 5 V.



Obrázek 4.5: OLED displej

4.4 Kapacitní tlačítka

Programovací modul je ovládán čtyřmi dotykovými tlačítky. Každé tlačítko obsahuje řídicí obvod TTP223 (viz. obr. 4.6), který zpracovává informace z dotykové plochy a převádí je na svůj výstup ve formě digitální informace vypnuto/zapnuto. Integrovaný obvod obsahuje automatickou kalibraci, díky které se při každém připojení napětí na tlačítka dojde k „přizpůsobení“ k danému prostředí a eliminují se nechtěné stisky způsobené materiálem krabičky nebo rušením z napájení. Vstupy integrovaného obvodu jsou velmi citlivé a je možné umístit dotykovou plochu až 15 mm od kapacitního tlačítka, pro kratší vzdálenosti je nutné doplnit kondenzátor C12 o velikosti 1 – 33 pF v závislosti na snímanou vzdálenost. Propojky SJ1 a SJ2 nastavují logickou hodnotu výstupu a definují funkci obvodu jako tlačítko nebo spínač. Tlačítka pracují s napájecím napětím 2,5 až 5,5 V. Jejich výhodou je velmi nízká spotřeba, proudový odběr se pohybuje mezi 2 až 3 uA. Po 10 sekundách neaktivity modul přechází automaticky do stavu nízkého odběru, ve kterém odběr dostane na 1,5 uA. Na obrázku 4.7 je finální rozmístění ovládacích prvků a jejich význam na krabičce programovacího modulu.



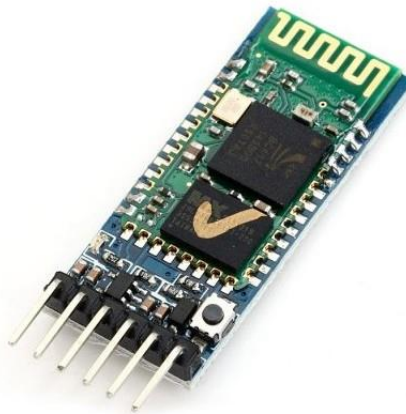
Obrázek 4.6: Schéma zapojení kapacitního tlačítka



Obrázek 4.7: Význam jednotlivých tlačítek

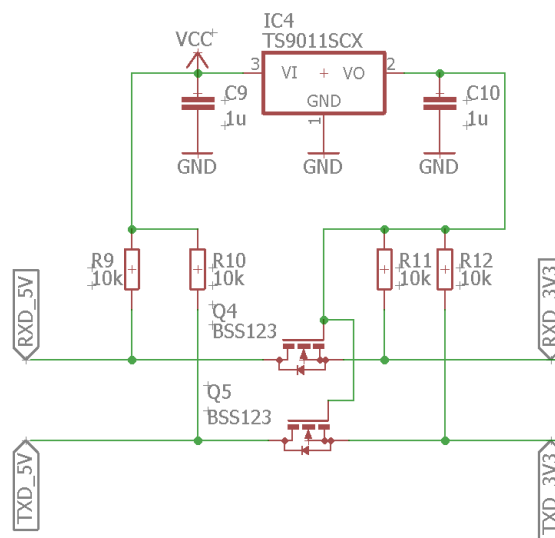
4.5 Bluetooth

Pro bezdrátovou komunikaci s programovacím modulem je využit komunikační Bluetooth modul HC-05 (viz. obr 4.8). Tento modul podporuje Bluetooth standard verze 2.0. Komunikace s mikrokontrolerem je zajištěna pomocí sériové linky (UART) s výchozí rychlostí 9600 baudů. Modul má výrobcem udávaný dosah okolo 10 metrů na volném prostranství. I když dosah modulu není nijak velký, pro tuto aplikaci je plně dostačující, protože při programování bude telefon v bezprostřední blízkosti programovacího modulu. Modul pracuje napájecího napětí 4 až 6 V s proudovou spotřebou 2 mA v klidovém režimu a 40 mA během komunikace nebo pokud není k němu žádné zařízení připojeno.



Obrázek 4.8: Bluetooth modul HC-05 (převzato z [9])

V případně nižšího napájecího napětí, než jsou 4 V, se modul automaticky vypne. Nevýhodou modulu je, že datové vstupy nevyužívají TTL logiky, ale 3,3 V logiky. Tento problém je řešen jednoduchým převodníkem úrovní, viz. obr 4.9. Základ tohoto převodníku tvoří stabilizátor na 3,3 V a dva tranzistory MOS-FET.

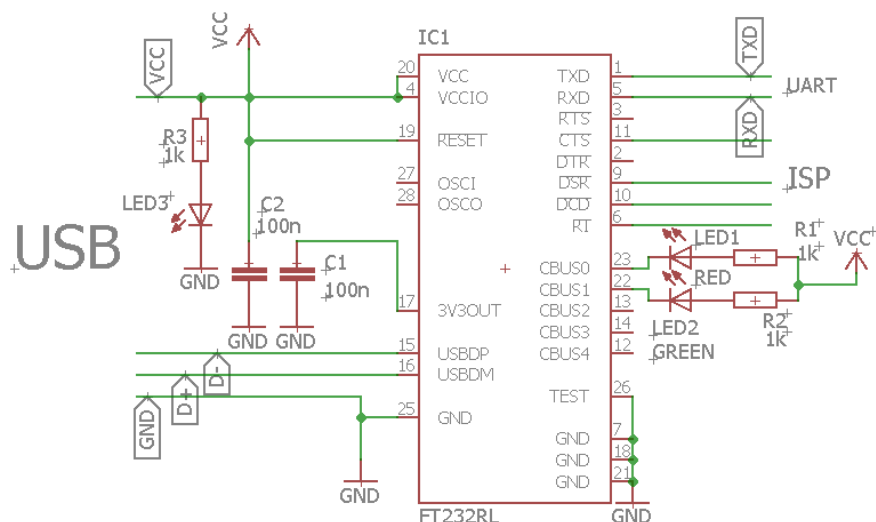


Obrázek 4.9: Převodník úrovní 5 V/ 3,3 V

Princip funkce obvodu je jednoduchý, pokud ani jedna strana nekomunikuje, pull-up rezistory způsobí, že je linka na obou stranách v logické jedničce. Rozdíl napětí mezi vývody „source“ a „gate“ tranzistoru je blízké nule a tranzistor je zavřen. Jakmile pravá strana (ta s nižším napětím) spojí linku se zemí (logická nula), rozdíl napětí mezi „source“ a „gate“ tranzistoru stoupne a tím se otevře. Logická nula se objeví i na pravé straně. Pokud levá strana spojí linku se zemí, dioda mezi source a „drain“ tranzistoru způsobí, že se zvětší rozdíl napětí mezi „source“ a „gate“ a tranzistor se otevře. Tím se na levé straně objeví logická nula.

4.6 USB/UART převodník

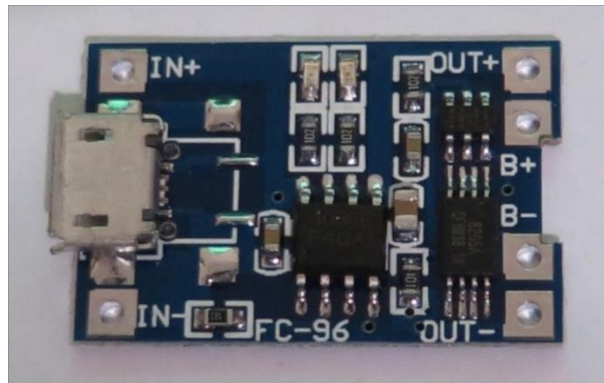
Srdcem USB/UART převodníku je obvod FT232RL, který v programovacím modulu zajišťuje dvě hlavní funkce. První a zároveň hlavní funkcí je podobně jako u Bluetooth modulu komunikace programovacího modulu s počítačem pomocí terminálových příkazů. Druhou funkcí je ISP rozhraní pro aktualizaci firmware programovacího modulu, bez použití externího programátoru. Díky tomu, že ISP rozhraní u FT232 funguje na principu „bit-bangování“, a není možné u něj nastavit přenosovou rychlost, která je s výchozím nastavením propojek procesoru příliš vysoká. Proto je pro oživení mikrokontroleru připraven konektor ISP rozhraní, přes který se nastaví propojky mikrokontroléru. Na obrázku 4.10 je schéma zapojení. Schéma obsahuje tři signalizační diody, první dvě signalizují přenos dat a třetí signalizuje připojení USB do počítače. Maximální rychlost přenosu dat přes převodník je 3000000 Baud.



Obrázek 4.10: Schéma zapojení UART/USB převodníku

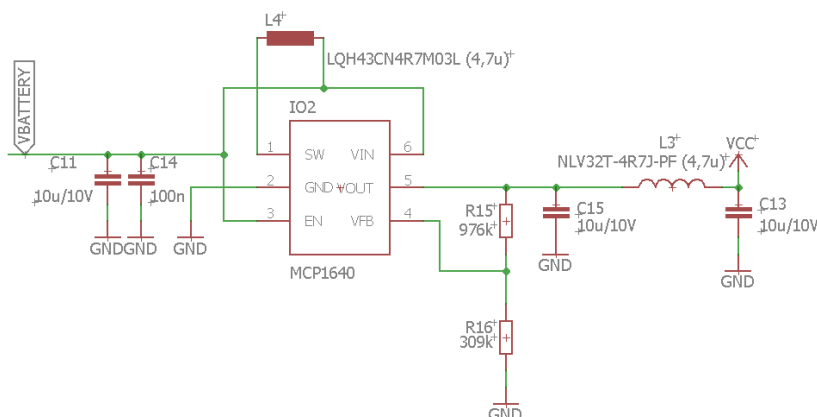
4.7 Napájení

Jednou z nejvíce klíčových částí je napájení programovacího modulu. Je nutné, aby zařízení bylo schopno pracovat jak z USB s napětím 5 V, tak i v terénu z Li-ion článku o napětí 3,7 V. Další omezujícím faktorem je napájení Bluetooth modulu, který vyžaduje napětí nejméně 4 V. Z těchto důvodů jsou napájecí obvody realizovány pomocí 5 V měniče a specializovaného nabíjecího modulu (viz. obr 4.11). Nabíjecí modul obsahuje mini USB nabíjecí konektor a je primárně určen pro power banky s jedním Li-ion článkem 18650 s nastavitelným nabíjecím proudem až do 1 A. Modul obsahuje dvě signalizační led diody, které signalizují informace o nabíjení a stavu nabití článku. Hlavním stavebním kamenem nabíječky je obvod TP4056. Jedná se o napěťový, resp. proudový zdroj určený pro nabíjení Li-ion článků. Hlavní výhodou tohoto obvodu je minimální množství externích součástek. Dále je modul vybaven obvodem DW01-P na ochranu baterie proti přebíjení nebo podbití. Horní mez, kdy je článek nabit je 4,21 V a ochrana proti podbití odpojuje baterii při napětí 2,54 V.



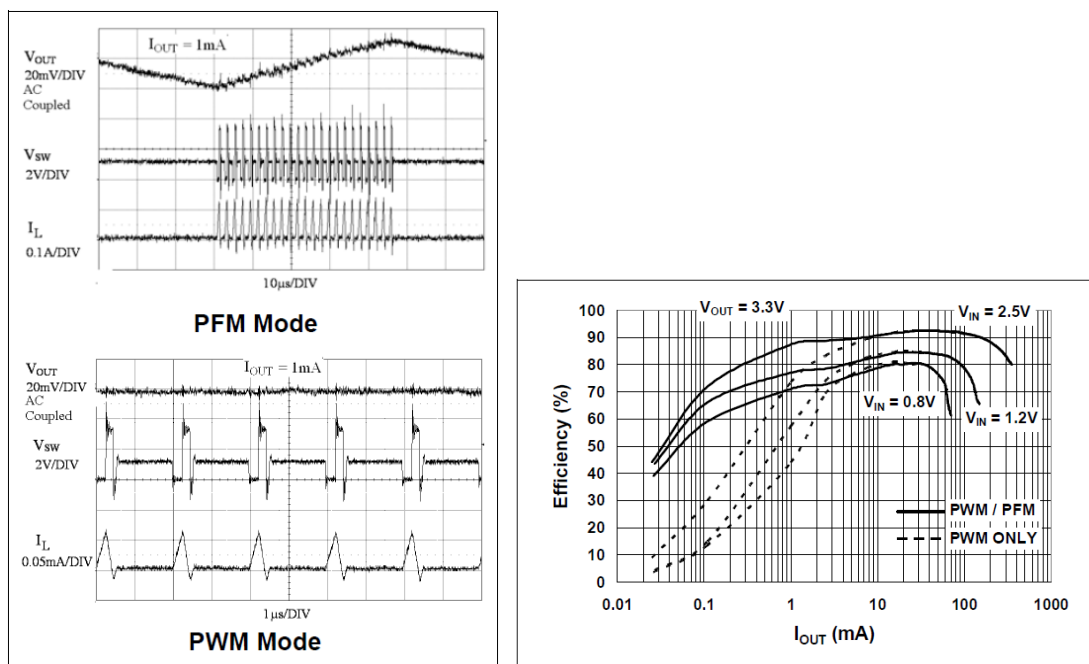
Obrázek 4.11: Napájecí modul (převzato z [10])

Měnič na 5 V je realizovaný integrovaným obvodem MCP1640 od společnosti Microchip (viz. obr 4.12). Jedná se o synchronní zvyšující měnič, který je speciálně určen pro bateriové aplikace.



Obrázek 4.12: Schéma měniče napětí

Velkou výhodou tohoto obvodu je schopnost pracovat již od napájecího napětí 0,35 V, vysoká účinnost a schopnost automatického přepínání mezi režimy PFM a PWM. Použití PFM režimu je velmi výhodné svojí vysokou účinností v době, kdy je programovací modul ve vypnutém stavu (a odebírá velmi nízký proud). Naopak pokud je programovací modul v zapnutém stavu a odebírá velký proud, využívá měnič PWM modulaci, která opět zajišťuje vysokou účinnost a nízké zvlnění výstupního napětí (viz. obr 4.13).

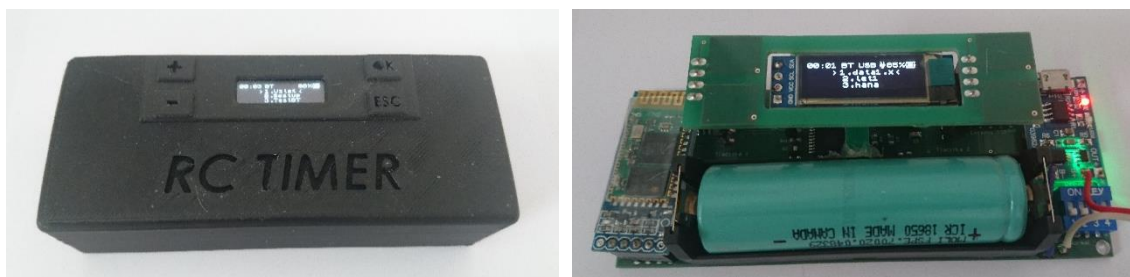


Obrázek 4.13: Srovnání PWM a PFM režimů (převzato z [12])

4.8 Praktická realizace prototypu programovacího modulu

Programovací modul tvoří sestava oboustranných desek plošných spojů o celkovém rozměru 112,7 x 43 x 28 mm. Výkresová dokumentace, včetně osazovacích výkresů, je umístěna v příloze C.

Programovací modul je kompletně osazen součástkami pouze v pouzdrech SMD. Sestava programovacího modulu je navržena tak, že ji lze namontovat do prototypové krabičky, jejíž dokumentace je v příloze G. Na obrázku 4.14 je finální prototyp programovacího modulu. V příloze D jsou detailní technické parametry finálního výrobku.



Obrázek 4.14: Finální prototyp programovacího modulu
(vlevo - v krabičce, vpravo - detail na sestavu)

5 ŘÍDÍCÍ SOFTWARE PROGRAMOVACÍHO MODULU

Firmware pro mikrokontroler programovacího modulu byl podobně jako pro modul časovače napsán v jazyce C ve vývojovém prostředí Atmel Studio.

5.1 Běh programu

V příloze F je kompletní vývojový diagram programu inicializace mikrokontroleru. Po připojení desky k napájecímu napětí dojde k automatické inicializaci procesoru, následně se inicializují sériová rozhraní, displej a nakonec vnitřní paměť. Po inicializaci displeje je zařízení schopno signalizovat chybové stavy na displeji, pokud nastane nějaká chyba hardwaru před tímto krokem jsou veškeré chybové stavy odesílány na obě sériová rozhraní (Bluetooth i UART/USB). V případě, že celá inicializace proběhne v pořádku, započne kontrola vnitřní paměti a pokud na pozici počtu uložených záznamů je nalezeno číslo v rozmezí 0 a 10, započne načtení odpovídajícího počtu záznamů a zobrazení na úvodní obrazovce. Systém organizace paměti funguje na principu ukazatelů uložených v první části paměti (viz. kapitola 4.2). Následně je program zacyklen v nekonečné smyčce, která periodicky obnovuje data na displeji na základě stisknutých tlačítek, udržuje systémový čas, obsluhuje komunikaci po sériových rozhraních a detekuje přítomnost čipu časovače.

Pokud program detekuje přítomnost čipu, je automaticky povolena uživateli možnost programování modulu časovače. Vlastní programování časovače je realizováno pomocí sběrnice I2C, přes kterou je nejdříve dotázán mikrokontroler na bezpečnostní klíč a pokud souhlasí je programování povoleno (nastaví WP pin do log. 0), následně je navázána komunikace přímo s pamětí EEPROM.

Hladký běh programu zaručuje implementace systému „watchdog“, který v jakékoliv nestandardní situaci (např. odpojení modulu časovače během kopírování dat) resetuje mikroprocesor.

5.2 Funkce programu

Hlavním stavebním prvkem časovače je mikrokontroler ATmega644P. Tento mikrokontroler má celou řadu periférií (viz. tab. 4.1). V tabulce 5.1 je výčet využitých periférií v řídicím programu a jejich účel použití.

Tabulka 5.1: Přehled využitých periférií mikrokontroleru

Název periferie	Využití
Timer/Counter0	Hlavní časovač pro modul <i>scheduler.c</i>
Timer/Counter1	Časovač pro maskování vstupů kapacitních tlačítek
A/D převodník - kanál 1	A/D převodník pro měření napětí baterie
UART0	Sériové rozhraní pro přenos dat s PC přes USB a debugování
UART1	Sériové rozhraní pro přenos dat se Smartphonem přes Bluetooth
ISP	Sériové programování procesoru
I2C	Sběrnice pro komunikaci s vnitřní EEPROM pamětí, displejem a EEPROM pamětí modulu časovače

V následujícím seznamu je uveden výčet všech důležitých funkcí, které časovač využívá. Díky relativní jednoduchosti nebyl program členěn do více modulů. Obsahuje pouze modul **main.c** a knihovny, které obsluhují jednotlivé periferie.

- **main()** – výchozí funkce programu. Nemá vstupní ani návratovou hodnotu. Po prvotní inicializaci periférií v ní dojde k zacyklení v nekonečné smyčce, ze které jsou volány funkce pro obsluhu OLED displeje - **barstav()**, sériových rozhraní - **CtiUART()**, napájení - **PWRmng()**, hodin - **hodiny()** a detekce čipu - **cip_detekce()**.
- **barstav()** – stěžejní funkce celého programu. Slouží k zobrazování obsahu na stavové liště hlavní obrazovky na displeji, dále obsahuje podfunkci **zobrazeni()**, která obsluhuje zobrazení obsahu hlavní obrazovky, všech menu i nastavení.
- **CtiUART()** – funkce, která kontroluje příjem nové zprávy z obou rozhraní UART, v případě detekce nové zprávy podfunkce **zpracujUART()** obslouží její rozklíčování a zpracování jejího obsahu.
- **PWRmng()** – funkce pro šetření energie, obsluhuje GPIO piny, které přes tranzistory zapínají displej a Bluetooth.
- **cip_detekce()** – funkce, která periodicky se pokouší navázat komunikaci s modulem časovače a vyhodnocuje jeho přítomnost.
- **hodiny()** – funkce, která pomocí neblokujícího čekání udržuje systémový čas
- **exteep_write_word()** – funkce pro zápis *uint16_t* čísla do vnitřní paměti

EEPROM (nebo do modulu časovače).

- *eeeprom_cti_u16()* - funkce pro čtení *uint16_t* čísla z vnitřní paměti EEPROM (nebo z modulu časovače)
- *eeeprom_adresa_zapis()* – funkce obsluhující generování ukazatelů na jednotlivé záznamy v paměti a jejich uložení do první sekce vnitřní paměti
- *eeeprom_smazat()* – funkce obsluhující mazání jednotlivých záznamů z paměti a formátování paměti (v případě poškození konzistence dat)
- *exportDat()* – funkce obsluhující odeslání vybraného záznamu po sériových rozhraních do počítače nebo mobilního telefonu
- *vypnout()* – funkce obsluhující vypnutí displeje, Bluetooth a zastavení všech procesů uvnitř mikroprocesoru vyjma hodin a kontroly ovládacích tlačítek

5.3 Zápis dat do vnitřní paměti

Obsluha paměti EEPROM je elementární prvkem celého programu, nicméně se jedná společně se čtením o jednu z nejvyužívanějších funkcí celého programu. Paměť je organizovaná ve formátu dvou bajtových čísel (viz. kapitola 4.2). Princip funkce následujícího zdrojového kódu je jednoduchá. Po zavolání funkce je nejdříve adresa i data rozdělena na dvě osmi bitové části. Následně je vytvořeno spojení s pamětí, zapsána data a ukončeno spojení. Nakonec se čeká po dobu ochranného intervalu, který je nutný, aby data v paměti byla korektně uložena. Princip zápisu je totožný jako zápis do modulu časovače.

```
void exteep_write_word(uint16_t addr, uint16_t data)
{
    uint8_t ladd = 0x00; //proměnná spodních 8 bitů adresy
    uint8_t hadd = 0x00; //proměnná horních 8 bitů adresy
    uint8_t ldata = 0x00; //proměnná spodních 8 bitů dat
    uint8_t hdata = 0x00; //proměnná horních 8 bitů dat
    hadd = addr >> 8; //bit. posun doprava (pro získání horních 8bitů)
    ladd = addr & 0xFF; //odstranění horních 8 bitů (uint16 -> uint8)
    hdata = data >> 8; //bit. posun doprava (pro získání horních 8bitů)
    ldata = data & 0xFF; //odstranění horních 8 bitů (uint16 -> uint8)
    i2c_start(EEPROM_I2C_ADDRESS_HIGH + I2C_WRITE); //start I2C
                                                komunikace, v režimu zápisu
    i2c_write(hadd); //zápis horních 8 bitů adresy
    i2c_write(ladd); //zápis dolních 8 bitů adresy
    i2c_write(hdata); //zápis horních 8 bitů dat
    i2c_write(ldata); //zápis dolních 8 bitů dat
    i2c_stop(); //ukončení I2C komunikace
    _delay_ms(20); //ochranný interval
}
```

5.4 Formátování dat pro uložení do vnitřní paměti

V tabulce 5.2 jsou uvedeny příkazy, kterými je možné do programovacího modulu zadávat nové časové sekvence. Porovnáním formátu příkazů a struktury vnitřní paměti plyne, že je nutné vstupní data přeskupit. Následující zdrojový kód popisuje pro lepší pochopení problematiky obsluhu formátování dat časové sekvence a uložení do paměti pro jeden kanál:

```
void zapis_eeprom(void)
{
    uint16_t EEPROM[255]; //pole dat pro uložení do EEPROM
    uint16_t adresa_zapis=0; //ukazatel adresy kam se začne zapisovat
    uint16_t maximum=0; //největší počet záznamů ze všech kanálů
    uint16_t pocitadlo=0; //proměnná pro uložení celkové délky sekvence
    uint16_t posun=0; //násobek pozice uložení v poli EEPROM
    uint8_t pom=0; //pomocná
    adresa_zapis=eeprom_adresa_zapis(true,0); //načtení adresy zápisu
    for(uint16_t i=0;lookupSl[i]!=0;i++) //formátování 1. kanálu
    {
        wdt_reset(); //reset watchdogu
        pocitadlo++; //inkrementace počítadla záznamů
        EEPROM[i+posun+1]=lookupSl[i]; //příprava výsledných dat
        if(pom==1) //podm. zaruč., že data délky trvání a střídý
            budou uloženy vedle sebe
        {
            posun=posun+4;
            pom=0;
        }
        else pom++;
    }
    if(pocitadlo>maximum) maximum=pocitadlo; //detekce nejdelšího kanálu
    pocitadlo=3*maximum; //vytvoření hranaté matice záznamů (neúplné
        kanály jsou doplněny nulami)
    EEPROM[0]=pocitadlo; //uložení na první pozici délku časové sekvence
    EEPROM[pocitadlo+1]=0;
    EEPROM[pocitadlo+2]=0xFFFF; //uložení na konec ochranných znaků
        (kontrolu konzistence paměti)
    for(uint16_t i=0;EEPROM[i]!=0xFFFF;i++) //zápis do paměti EEPROM
    {
        wdt_reset(); //reset watchdog
        exteep_write_word(adresa_zapis+(i*2),EEPROM[i]);
    }
    pocet_zaznamu=eeprom_cti_ul6(EEPROM_POCET_ZAZNAMU); //aktualizace
        počtu záznamů v RAM
    eeprom_jmeno_zapis(pocet_zaznamu); //uložení jména sekvence
    eeprom_adresa_zapis(false,pocitadlo); //aktualizace ukazatelů paměti
    pocet_zaznamu=eeprom_cti_ul6(EEPROM_POCET_ZAZNAMU); //aktualizace
        počtu záznamů v RAM
    cteni_eeprom_jmena(); //aktualizace seznamu jmen v RAM
    wdt_reset(); //reset watchdog
}
```

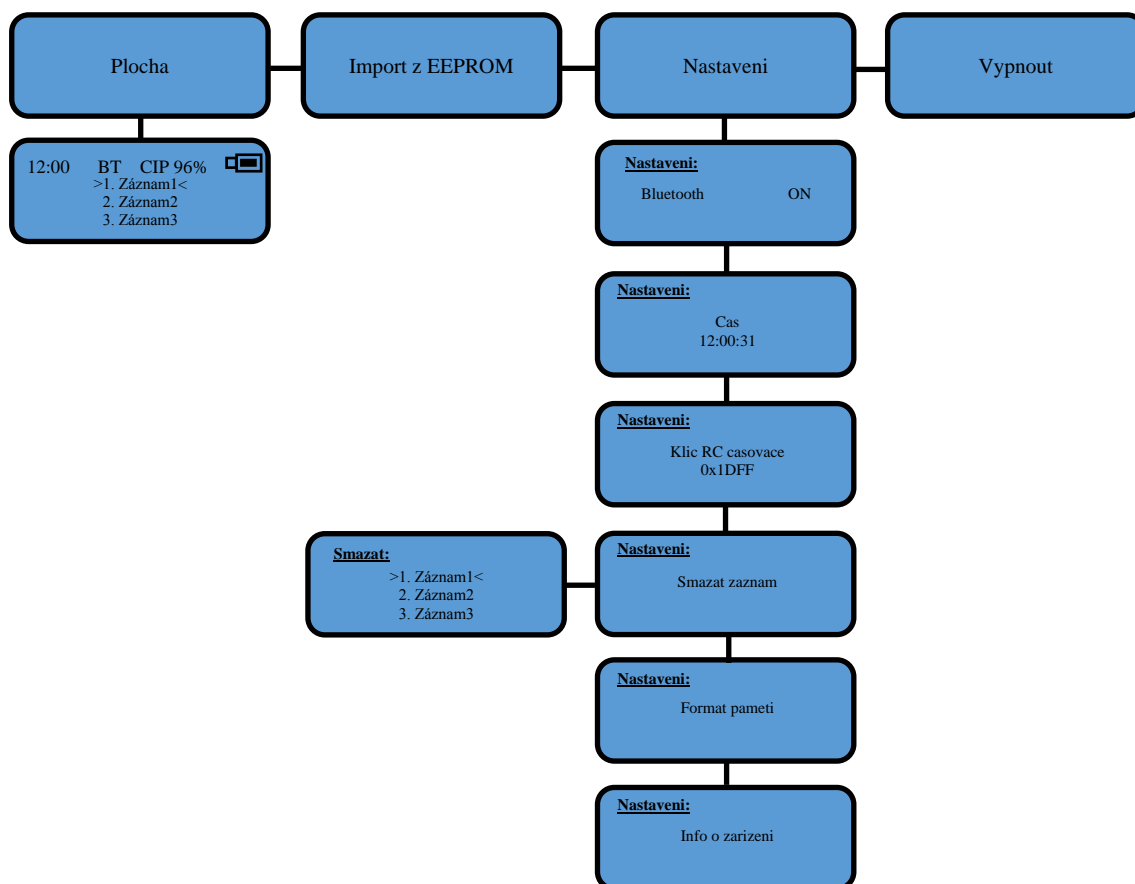
5.5 Příjem dat po sériových linkách

Pro bezchybný příjem příkazů po sériových linkách byl vytvořen následující zdrojový kód. Obsluha sériových rozhraní UART0 i UART1 je totožná. Při každém přijatém znaku nastává přerušení, ve kterém je obslouženo sestavení jednotlivých znaků do celistvé zprávy. Toto řešení má výhodu v tom, že programovací modul může přijmou najednou celý řetězec příkazu, ale vyrovná se i s jednotlivými znaky přicházejícími delší dobu po sobě, tak jak je uživatel píše do terminálové aplikace.

```
ISR(USART0_RX_vect) {
    char prijaty_znak;
    static uint8_t i = 0; // počítadlo znaku v přijímané zprávě
    prijaty_znak = UDR0; //překopírování přijatého znaku z bufferu UARTu
    if(prijaty_znak==' ') prijaty_znak='-'; //mezery nahrazují pomlčky
    if((prijaty_znak != '\n') && (prijaty_znak != '\r') &&
(i<DELKA_ZPRAVY-2)){ //pokud není konec zprávy nebo není ještě moc dlouhá
        docasne[i] = prijaty_znak; // uložíme nově příchozí znak
        i++; //inkrementace počítadla
    }
    else
    {
        docasne[i] = '\0'; // uložíme na konec znak \0 - konec řetězce
        strncpy(zprava,docasne,DELKA_ZPRAVY);
        nova_zprava = 1; // signalizace programu, že má novou zprávu
        i = 0; // příprava na příjem nové zprávy
    }
}
```

5.6 Ovládání firmware programovacího modulu

Uživatelsky pohodlné ovládání programovacího modulu zajišťuje grafický displej s jednoduchým textovým rozhraním. Na obrázku 5.1 je mapa ovládacího rozhraní, ze které je patrné, jak se dostat k jednotlivým funkcím programovacího modulu.



Obrázek 5.1: Mapa ovládacího rozhraní programovacího modulu

5.6.1 Plocha

Výchozí obrazovka celého systému je pod obrazovka položky v menu *Plocha*. Na tuto obrazovku se vždy uživatel dostane po zapnutí zařízení a je zní možné odesílat časové sekvence do časového modulu. Na prvním řádku displeje se zobrazuje stavová lišta, která zobrazuje:

- aktuální systémový čas
- zkratku BT, pokud je zapnutý Bluetooth
- slovo CIP, pokud je detekováno připojení modulu časovače
- aktuální stav baterie v procentech
- grafické vyjádření stavu baterie, včetně indikace vybité baterie

Dále obrazovka obsahuje až deset časových sekvencí, zobrazených na trojstránkovém seznamu. V případě, že je připojen modul časovače, je ihned po stisku tlačítka přepočítávána vybraná sekvence do časovače. V případě, že modul časovače je neočekávaně odpojen během přenosu dat, dojde k automatickému restartu programovacího modulu a přenesená data uložená v modulu časovače budou nekonzistentní (po připojení do letadla nedojde k rozsvícení zelené led diody a let nepůjde uskutečnit).

5.6.2 Import z EEPROM

Nabídka *Import z EEPROM* umožňuje zkopírovat sekvenci dat z modulu časovače do vnitřní paměti programovacího modulu. Stiskem tlačítka **OK** je zahájeno kopírování dat. Výsledná sekvence dat, která se uloží do vnitřní paměti časovače dostane název *ImportEE*. Pokud modul časovače není připojen, kopírování dat neproběhne. V případě, že modul časovače je neočekávaně odpojen během přenosu dat, dojde k automatickému restartu programovacího modulu a přenesená data jsou zahozena.

5.6.3 Nastavení

Nabídka *Nastavení* umožňuje uživateli měnit veškerá nastavení programovacího modulu. Menu obsahuje šest položek:

- *Bluetooth* – nastavením lze rozhraní zapnout nebo vypnout. Změna nastavení se provádí pomocí tlačítka **OK**. Pokud je Bluetooth vypnutém stavu klesne celková spotřeba energie o 50 %. Dále z hlavní obrazovky zmizí zkratka BT. Před dalším bezdrátovým programováním je nutné opět Bluetooth zapnout.
- *Cas* – nastavením lze měnit ručně systémový čas. Čas lze zadávat ve 24 hodinovém formátu. Zadávání proběhne stiskem tlačítka **OK** – dojde k potvrzení části hodin, následně lze pomocí + a – hodiny měnit. Po opětovném stisku tlačítka **OK** dojde k potvrzení minutové části a nastavení se provede obdobně, potvrzení se provede opětovným stiskem **OK**. Při změně času (letní na zimní nebo opačně) je nutná ruční změna.
- *Klic RC casovace* – nastavením lze měnit bezpečnostní klíč modulu časovače. Pokud klíč nesouhlasí s naprogramovaným klíčem v modulu časovače, není možné časovač programovat. Nastavení se provádí stiskem tlačítka **OK** a následovným stiskem tlačítek + a -, potvrzení se provede opětovným stiskem **OK**.
- *Smazat zaznam* – mazání jednotlivých časových sekvencí ze seznamu. Mazání se provede stiskem tlačítka **OK**, následně dojde ke zobrazení všech uložených časových sekvencí, pomocí + a – lze vybrat požadovaný záznam a následným opětovným stiskem **OK** se provede mazání. Jedná se o časově náročnější operaci, protože dochází k rozsáhlým přesunům dat ve vnitřní paměti. Během této operace nesmí být od programovacího modulu odpojeno napájení, jinak dojde k nenávratnému poškození integrity dat ve vnitřní paměti.
- *Format pameti* – formátování paměti slouží k opravě (k celkovému vymazání) paměti, pokud dojde k poškození integrity vnitřní paměti programovacího modulu. Formátování se provede dvojitým stiskem tlačítka **OK**. Jedná se o operaci, která může trvat i 5 minut.
- *Info o zarizeni* – zobrazení informací o zařízení, o jeho verzi hardware, firmware a informací o autorovi.

5.6.4 Vypnutí

Režim vypnutí přepne zařízení do úsporného režimu. Dojde k vypnutí displeje, Bluetooth modulu a zastavení veškerých procesů v MCU (vyjma běhu systémových hodin). Opětovné probuzení se provede dlouhým stiskem tlačítka **OK**. Pro dlouhodobé vypnutí je doporučeno použít k vypnutí hlavního vypínače ze zadní strany programovacího modulu.

5.6.5 Pohyb v nabídkách programovacího modulu

Pohyb v jednotlivých menu je realizováno pomocí kapacitních tlačítek (viz. kapitola 4.4, a obr. 4.7.). Význam jednotlivých tlačítek:

- + slouží k posouvání v nabídkách nahoru a doprava
- - slouží k posouvání v nabídkách dolů a doleva
- **OK** slouží ke stupu do podmenu a potvrzování
- **ESC** slouží k výstupu z podmenu a rušení

5.7 Komunikační příkazy pro programovací modul

Pro ovládání a programování programovacího modulu je vytvořen jednoduchý ovládací jazyk, který tvoří série příkazů. Pomocí těchto příkazů lze zařízení ovládat jak pomocí rozhraní Bluetooth, tak pomocí rozhraní USB. V tabulce 5.2 jsou uvedeny všechny příkazy se kterými programovací modul umí pracovat.

Tabulka 5.2: Přehled komunikačních příkazů pro ovládání programovacího modulu

Formát příkazu	Význam příkazu	Další informace k syntaxi
INFO	Zobrazení informací o verzi softwaru, hardwaru a autorovi	-
SET TIME XX:XX END	Nastavení systémového času	Na místo XX:XX se zadá čas ve 24 hodinovém formátu
SET NAME YYYY S1 A1,A2,AN1,AN2. S2 B1,B2,BN1,BN2. SN C1,C2,CN1,CN2. END	Zadání nového záznamu	Na místo YYYY se zadává jméno záznamu, které bude zobrazeno na displeji (max. 7 znaků). Za S1 následuje sekvence dat pro regulátor, za S2 a S3 následují sekvence pro jednotlivá serva. Na místo A1, A2 nebo B1, B2 nebo C1, C2 se zadává časový záznam ve formátu A1 – délka trvání záznamu [x100 ms], A2 – střída záznamu [x10 μs].
EXPORT X END	Stažení uloženého záznamu z vnitřní paměti	Na místo X se zadá číslo záznamu (1-10), který je požadováno exportovat

Velikost písmen příkazů je libovolná. Programovací modul na každý z příkazů odpovídá zpětným odesláním příkazu, ve kterém jsou mezery nahrazené pomlčkami, dále je řetězec následován odpovědí v závislosti na povahu příkazu. Nakonec je celá odpověď zakončena řetězcem CTENI OK.

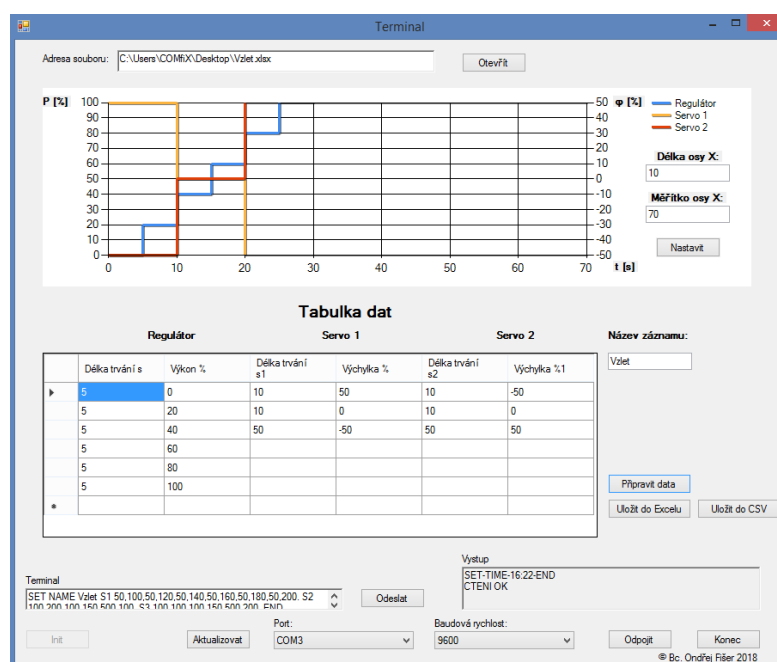
Implementace jednoduchých komunikačních příkazů má hlavní výhodu v tom, že díky jednoduchosti může uživatel naprogramovat časové sekvence do programovacího

modulu přes Hyperterminal (který je součástí Windows) i bez jakéhokoliv dalšího ovládacího softwaru.

6 OVLÁDACÍ APLIKACE PRO WINDOWS

Pro pohodlné vytváření a úpravu časových sekvencí byla vytvořena aplikace pro operační systém Windows s uživatelským prostředím, které umožňuje intuitivní obsluhu a přenos dat do programovacího modulu (viz. obr. 6.1). Aplikace byla napsána v jazyce Visual Basic s využitím vývojového prostředí Microsoft Visual Studio 2015. Instalátor programu včetně zdrojových kódů je na přiloženém CD.

Aplikace je optimalizována pro minimální rozlišení 1024 x 768. Pro navázání komunikace s programovacím modulem je nutné v nabídce pod označením *Port*: zvolit číslo sériového portu, které bylo přiřazeno USB/UART v programovacím modulu po připojení do USB portu počítače. V případě, že bylo zařízení připojeno až po zapnutí aplikace, je třeba stisknout tlačítko *Aktualizovat*. Dále zvolit baudovou rychlost, která je ve výchozím nastavení 9600 Baud a nakonec stisknout tlačítko *Init*. Po úspěšném spojení je aktualizován systémový čas programovacího modulu.



Obrázek 6.1: Aplikace Terminal pro MS Windows

V prostřední části aplikace, se nachází zdrojová tabulka dat. Na základě této tabulky se vykresluje grafická vizualizace ve formě grafu. Velkou výhodou aplikace je, že její jádro je postaveno na Microsoft Access Database Engine a je plně kompatibilní se soubory programu Microsoft Excel. Díky tomu je nejen možné načítat soubory ve formátu *.xlsx*, ale i editovat a ukládat. Aplikace spolehlivě funguje i na počítačích, na kterých není instalován balík MS Office. Ukládání dat z aplikace je možné, jak už bylo zmíněno ve formátu *.xlsx* nebo ve formátu *.csv*. Druhý formát je určený pro ovládací aplikaci pro chytré telefony běžící na platformě Android.

Tabulka dat je ošetřena proti chybám nekonzistence (např. prázdné buňky nebo hodnoty mimo rozsah). Pro odeslání dat do programovacího modulu stačí po vyplnění tabulky a jména záznamu kliknout na tlačítko *Připravit data* a následně na tlačítko *Odeslat*. Následně jsou data odeslána do zařízení.

Jak bylo zmíněno výše, s mikroprocesorem program komunikuje skrze USB/UART převodník (viz. kapitola 4.6). Z programátorského hlediska se k tomu přistupuje jako ke komunikaci po sériové lince. V následující části zdrojového kódu je otevření sériové linky se zařízením a nastavení místního času.

```
Private Sub initButton_Click(sender As Object, e As EventArgs) Handles
initButton.Click

    Dim output As String 'proměnná výstupního zásobníku
    output = "SET TIME " + Mid(TimeString, 1, 5) + " END" 'uložení do
                                     zásobníku nastavení aktuálního času
    SerialPort1.PortName = portComboBox.Text 'načtení čísla portu
    SerialPort1.BaudRate = baudComboBox.Text 'načtení baudové
                                     rychlosti
    SerialPort1.Open() 'otevření sériové linky
    SerialPort1.Write(output & vbCr) 'zápis času na seriový port

    initButton.Enabled = False 'zakázání tlačítka Init
    writeButton.Enabled = True 'povolení tlačítka Odeslat
    closeButton.Enabled = True 'povolení tlačítka Odpojit

End Sub
```

V další části zdrojového kódu je popsáno otevření libovolného excelového dokumentu pomocí Microsoft Access Database Engine. Engine není běžnou součástí Windows, proto je přiložen na CD a je ho nutné pro správnou funkci nainstalovat.

```
Private Sub openButton_Click(sender As Object, e As EventArgs) Handles
openButton.Click
    OpenFileDialog1.ShowDialog() 'zobrazení dialogu Otevřít
    addressBox.Text = OpenFileDialog1.FileName 'zobrazení adresy
                                     v políčku adresa

    Try
        Dim MyConnection As System.Data.OleDb.OleDbConnection
        Dim dataSet As System.Data.DataSet
        Dim MyCommand As System.Data.OleDb.OleDbDataAdapter
        Dim path As String = OpenFileDialog1.FileName 'inicializace
                                     proměnných

        MyConnection = New
        System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB
.12.0;Data Source=" + path + ";Extended Properties=Excel
12.0;")
        MyCommand = New System.Data.OleDb.OleDbDataAdapter("select *
from [List1$]", MyConnection) 'inicializace Database engine

        dataSet = New System.Data.DataSet
        MyCommand.Fill(dataSet) 'načtení tabulky do proměnné dataSet
        DataView.DataSource = dataSet.Tables(0) 'zobrazení tabulky v
                                     programu

        MyConnection.Close() 'zavření souboru
    Catch ex As Exception 'v případě jakékoliv chyby zobrazení chyby
        MsgBox(ex.Message.ToString)
    End Try
```



```

If addressBox.Text <> Nothing Then 'zobrazení jména souboru
                                v políčku jména sekvence dat
    nameTextBox.Text =
Mid(System.IO.Path.GetFileName(OpenFileDialog1.FileName), 1, 7)
End If
End Sub

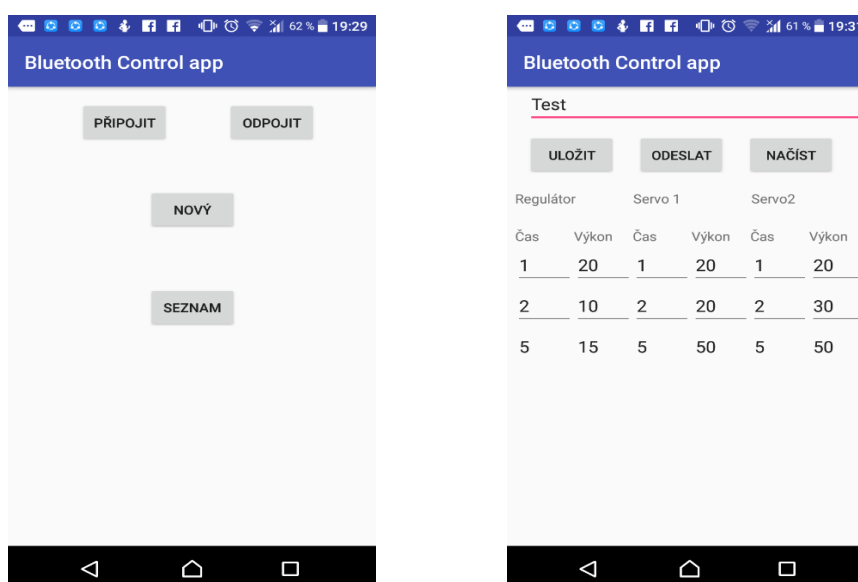
```

Implementace předchozího kusu kódu má dvě velké výhody v tom, že zdrojový soubor otevře jen na malou chvilku, načte z něj data a opět ho uvolní pro práci dalších programů, tzn. že souběžně s aplikací může být tabulka otevřená např. i v Excelu, kde může být souběžně editována. Druhou výhodou je použití metody *Try*, díky které program nespadne při chybě, ale vypíše chybové hlášení a pokračuje dál. Kompletní zdrojové kódy jsou obsahem přílohy na CD.

7 OVLÁDACÍ APLIKACE PRO ANDROID

Jedním z požadavků pro realizaci programovacího modulu byla možnost přenosu dat do zařízení pomocí chytrého telefonu. Pro pohodlné vytváření a úpravu časových sekvencí v terénu byla vytvořena aplikace pro platformu Android s uživatelským příjemným prostředím. Aplikace byla napsána v jazyce Java s využitím vývojového prostředí Android Studio. Balíček aplikace včetně zdrojových kódů je na přiloženém CD.

Aplikace je plně kompatibilní se systémem Android od verze 4.0 až po 7.1. Je optimalizovaná pro všechna rozlišení, které mají komerčně prodávané telefony. Na obrázku 7.1 jsou zobrazeny dvě hlavní obrazovky. Pro připojení k zařízení je nutné se se zařízením nejdříve spárovat (v nastavení Bluetooth), následně po kliknutí na tlačítko *PŘIPOJIT* se zobrazí seznam spárovaných zařízení. Po klepnutí na zařízení RC TIMER dojde k automatickému připojení.



Obrázek 7.1: Aplikace Bluetooth Control app pro Android
(vlevo – hlavní obrazovka, vpravo – editace záznamů)

Nová časová sekvence se vytvoří kliknutím na tlačítko *NOVÝ*. Dále je systém zadávání obdobný jako u aplikace pro MS Windows. Tabulka dat je podobně jako u aplikace pro Windows ošetřena proti chybám nekonzistence (např. prázdné buňky nebo hodnoty mimo rozsah). Kdykoliv během editace dat je možné je uložit stisknutím tlačítka *ULOŽIT*. Data se uloží ve formátu *.csv* do interní paměti zařízení: */storage/emulated/0/Bluetooth Control App/*. Podobně jako ukládání je možné data odeslat tlačítkem *Odeslat*. Na hlavní obrazovce se pod tlačítkem *SEZNAM* skrývá seznam všech uložených souborů. Soubory časovými sekvencemi je možné vytvářet i pomocí aplikace Terminal na počítači (viz. kapitola 6) a pomocí datového kabelu je uložit do telefonu.

Z programátorského hlediska je největší problém u vývoje aplikací pro Android kompatibilita mezi jednotlivými verzemi systému Android. Protože starší verze této platformy mají některé funkce, které se ve vyšších verzích Androidu z důvodu zastarání již nevyskytují nebo jsou nahrazeny jinými. Dobrým příkladem je například zabezpečení zápisu do interního úložiště, pokud tedy chce aplikace ukládat data do interního úložiště, musí požádat jádro systému o jeho povolení. Na základě uživatele je následně povoleno nebo zamítnuto. Tato funkce je ve všech verzích Androidu od verze API 23 (tj. od Androidu verze 6.0). Následující část zdrojového kódu nastiňuje řešení problému.

```
public boolean Save_to_file() //funkce uložení do souboru
{
    try{
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)
!= PackageManager.PERMISSION_GRANTED) { //kontrola povolení k zápisu
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) { //žádost o povolení
                else {
                    ActivityCompat.requestPermissions(this, //žádost o
                        povolení není nutná, pro nižší verze API
                        new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},0);
                }
            } else {
                //povolení bylo schváleno
            }
            if(!isExternalStorageWritable()) {
                throw new IOException(); //zobrazení chybového hlášení
            }
        }
    }
}
```

8 ZÁVĚR

V práci byl proveden návrh koncepce obvodového zapojení a návrh firmwaru časovače pro RC modely používaných v leteckém modelářství. Navržený časovač je schopný řídit modely bez rádiového řízení s běžnými analogovými servy a regulátory. Data se záznamy časových úseků jsou uloženy v EEPROM paměti a je možné je externě přepisovat přes sběrnici I2C. Návrh dále obsahuje dvě signalizační LED, které indikují aktuální stav časovače nebo případnou chybu paměti. Byl realizován funkční prototyp časovače, jeho funkčnost je možné demonstrovat na zkušebním standu, který byl postaven během vývoje.

Dále byl navržen programovací modul, pro snadné programování modulu časovače. Do vnitřní paměti lze uložit až deset časových sekvencí, které lze jednoduše nahrát do modulu časovače pouhým připojením a stiskem několika tlačítek. Ovládání zajišťuje grafický OLED displej a čtyři kapacitní tlačítka. Komunikovat s okolím a ukládat do zařízení nové časové sekvence je možno prostřednictvím USB a Bluetooth rozhraní. Programovací modul je plně mobilní, díky vestavěné Li-ion baterii. Během vývoje byl realizován funkční prototyp programovacího modulu a k němu potřebné softwarové vybavení pro snadné vytváření časových sekvencí jak pro počítače se systémem Windows, tak pro chytré telefony se systémem Android.

LITERATURA

- [1] LES SERVOS NUMERIQUES, DIT AUSSI » DIGITAUX » [MATERIEL]. [online]. 2016 [cit. 2017-12-05]. Dostupné z: <http://www.aeromodelisme.org/site/les-servos-numeriques-dit-aussi-digitaux-materiel>.
- [2] RC přijímač Hitec Minima 6T – bez telemetrie. [online]. 2017 [cit. 2017-12-05]. Dostupné z: <http://www.modelarina.cz/rc-prijimac-hitec-minima-6t-bez-telemetrie-p-2255.html>.
- [3] Jak to dát celé dohromady – podruhé. [online]. 2004 [cit. 2017-12-05]. Dostupné z: <http://www.mo-na-ko.net/lode-sestava2.htm>.
- [4] Beginners' Guide. [online]. [cit. 2017-12-05]. Dostupné z: <http://rcplanes.000webhostapp.com/guide1.htm>.
- [5] Valkyrie – časovač. [online] Promodels 2017 [cit. 2017-12-05]. Dostupné z: <http://www.promodels.cz/cz-detail-907265-valkyrie-casovac.html>.
- [6] 24LC64 [online]. Microchip 2002: 64 K I2C™ Serial EEPROM [cit. 2017-12-05]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/21189f.pdf>.
- [7] ATmega8A-MU [online]. Microchip 2015: 8 - bit AVR Microcontroller [cit. 2017-12-05]. Dostupné z: http://www.atmel.com/images/atmel-8159-8-bit-avr-microcontroller-atmega8a_datasheet.pdf.
- [8] Atmel Studio 7 [online]. Atmel 2017: Easier to Use and More Powerful Than Ever [cit. 2017-12-05]. Dostupné z: <http://www.atmel.com/microsite/atmel-studio/>
- [9] Arduino Bluetooth modul HC-05 [online]. 2017 [cit. 2017-12-05]. Dostupné z: <http://navody.arduino-shop.cz/navody-k-produktum/arduino-bluetooth-modul-hc-05.html>
- [10] TP4056 [online]. HAOYU Electronics 2017: Micro USB 5 V 1 A Lithium Battery Charger with Protection [cit. 2017-12-05]. Dostupné z: <http://www.hotmcu.com/tp4056-micro-usb-5v-1a-lithium-battery-charger-with-protection-p-176.html>
- [11] POVALAČ, A. *Počítačové systémy a jejich aplikace, přednášky*. Vyd. 1. V Brně: Vysoké učení technické, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky.
- [12] HW.cz [online]. Navrhujeme zvyšující měnič, pracující z jediného článku. [cit. 2018-05-06]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/navrhujeme-zvysujici-menic-pracujici-z-jedineho-clanku-1-dil.html>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

BEC	Battery eliminating circuit, stabilizátor napětí.
RC	Radio controlled, rádiem řízený.
PWM	Pulse-width modulation, pulzně šířková modulace.
VF	Vysokofrekvenční.
ISM	Industrial, scientific and medical band, pásmo využívají se k vysílání v průmyslu.
I2C	I-squared-C, dvoudrátové rozhraní pro komunikaci mezi mikrokontrolerem a okolními periferiemi.
WP	Write protect, ochrana zápisu.
ISP	In-system programming, programovací rozhraní mikrokontroleru.
LED	Light-Emitting Diode, dioda emitující světlo.
MCU	Microcontroller unit, mikroprocesorová jednotka.
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektronicky vymazatelná paměť pouze pro čtení.
RAM	Random-access memory, polovodičová paměť s přímým přístupem umožňující čtení i zápis.
RTOS	Real Time Operating Systém, operační systém reálného času.
USB	Universal Serial Bus (USB) je univerzální sériová sběrnice, moderní způsob připojení periférií k počítači.
UART	Universal Asynchronous Receiver and Transmitter, rozhraní pro sériovou komunikaci v asynchronním režimu.
SMD	Surface Mount Device, součástky určené pro povrchovou montáž.
SPI	Serial Peripheral Interface, sériové periferní rozhraní pro komunikaci s periferiemi mikrokontroleru.
DPS	Deska Plošných Spojů.
AVR	8bitový RISC mikropočítač od Microchip.
OLED	Organic light-emitting diode, typ LED diod, kde se jako elektroluminiscenční látka využívají organické materiály
PFM	Pulse-frequency modulation, Impulsní frekvenční modulace
t	čas [s]
φ	úhel natočení [°]
U	napětí [V]
I	proud [A]

V_{CC}	vstupní napětí
GND	zem
R	odpor [Ω]

SEZNAM PŘÍLOH

A	LETOVÉ SCÉNÁŘE	47
A.1	Letový scénář INDOOR	47
A.2	Letový scénář OUTDOOR	48
B	VÝKRESOVÁ DOKUMENTACE ČASOVAČE PRO RC MODELY	50
B.1	Schéma zapojení modulu časovače.....	50
B.2	Deska plošného spoje časovače – top (strana spojů)	50
B.3	Deska plošného spoje časovače – bottom (strana spojů).....	51
B.4	Osazovací výkres desky plošného spoje časovače – top	51
B.5	Osazovací výkres desky plošného spoje časovače – bottom	51
C	VÝKRESOVÁ DOKUMENTACE PROGRAMOVACÍHO MODULU	52
C.1	Schéma zapojení procesorové desky programovacího modulu	52
C.2	Deska plošného spoje procesorové desky prog. modulu – top	53
C.3	Deska plošného spoje procesorové desky prog. modulu – bottom.....	53
C.4	Osazovací výkres desky plošného spoje procesorové desky – top	54
C.5	Schéma zapojení kapacitních tlačítek programovacího modulu.....	54
C.6	Deska plošného spoje tlačítek programovacího modulu – top	55
C.7	Deska plošného spoje tlačítek programovacího modulu – bottom	55
C.8	Osazovací výkres desky plošného spoje tlačítek – bottom.....	55
C.9	Schéma zapojení propojovací desky programovacího modulu	56
C.10	Deska plošného spoje propojovací desky prog. modulu – top	56
C.11	Deska plošného spoje propojovací desky prog. modulu – bottom	56
D	SEZNAMY POUŽITÝCH SOUČÁSTEK	57
D.1	Seznam součástek modulu časovače.....	57
D.2	Seznam součástek programovacího modulu	57
E	TECHNICKÉ PARAMETRY	59
E.1	Časovač pro RC modely	59
E.2	Programovací modul.....	59
F	VÝVOJOVÝ DIAGRAM INICIALIZACE PROG. MODULU	60
G	VÝKRESOVÁ DOKUMENTACE ÚLONÉHO BOXU	61
G.1	Úložný box – hlavní část	61

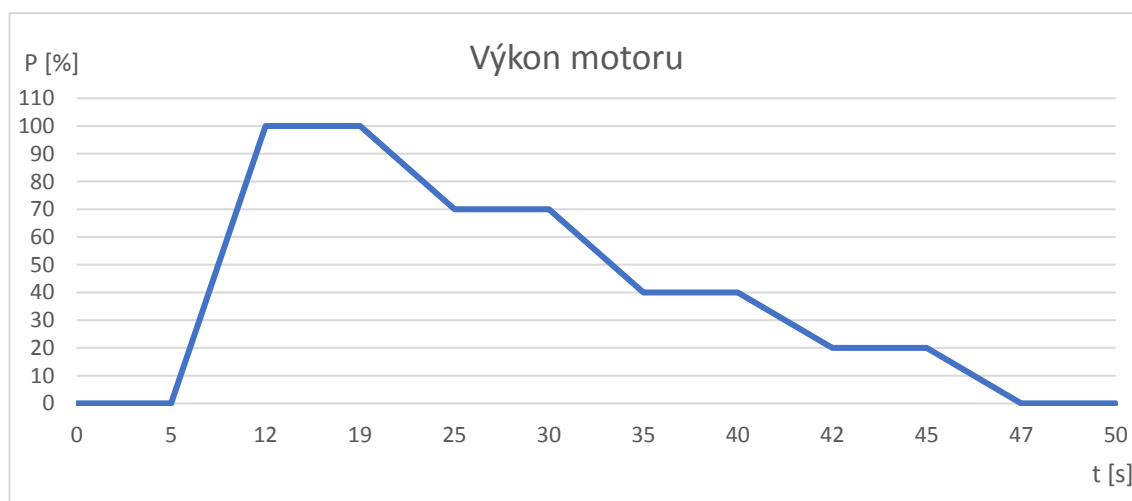
G.2	Úložný box – panel	62
G.3	Úložný box – zadní kryt.....	63
G.4	Úložný box – záslepka.....	64
G.5	Úložný box – sestava	65

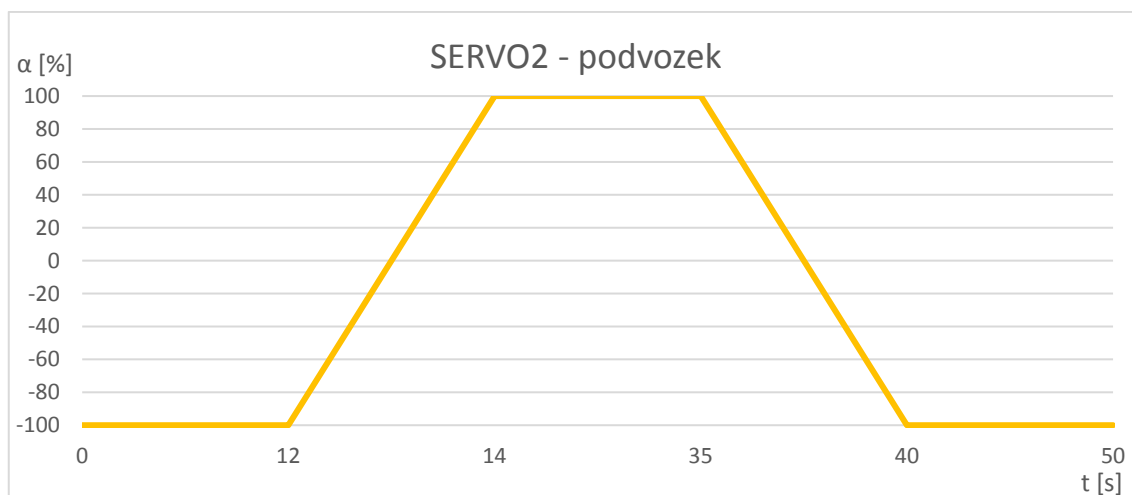
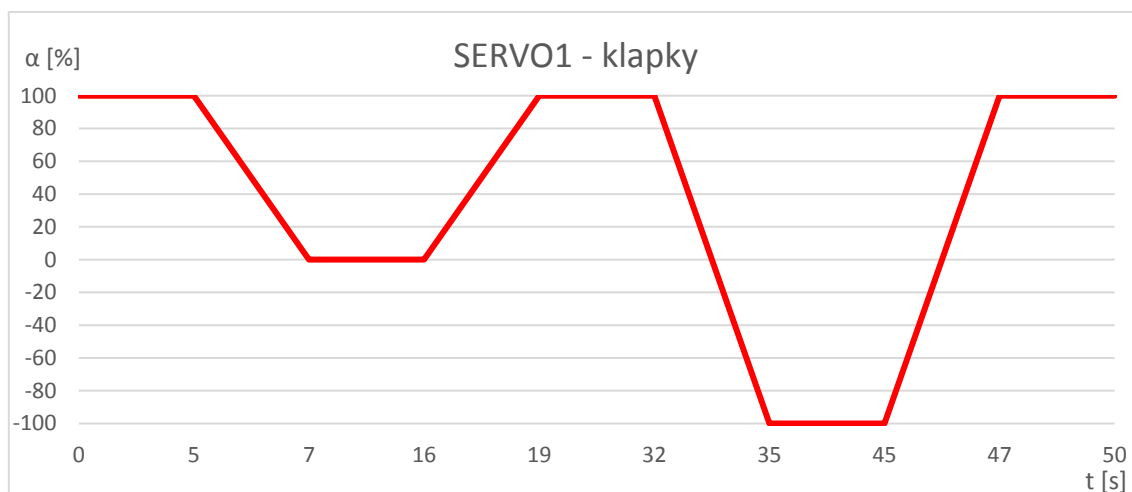
A LETOVÉ SCÉNÁŘE

A.1 Letový scénář INDOOR

Zmáčknutí
startovacích
o tlačítka
TL.

Tabulka hodnot serv a regulátoru během letu			
t [s]	Výkon motoru [%]	SERVO1 klapky [%]	SERVO2 podvozek [%]
0	0	100	-100
5	0	100	
7		0	
12	100		-100
14			100
16		0	
19	100	100	
25	70		
30	70		
32		100	
35	40	-100	100
40	40		-100
42	20		
45	20	-100	
47	0	100	
50	0	100	-100



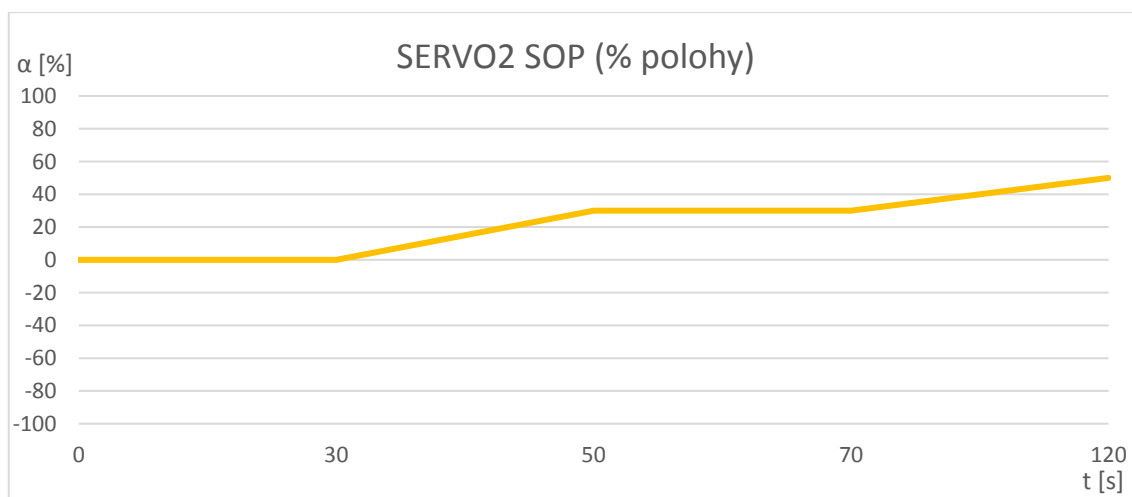
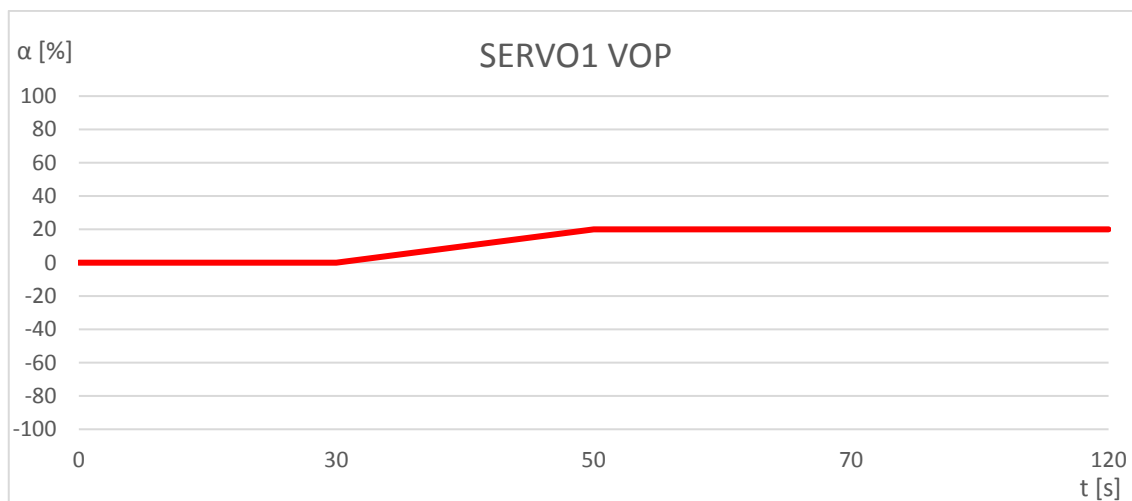
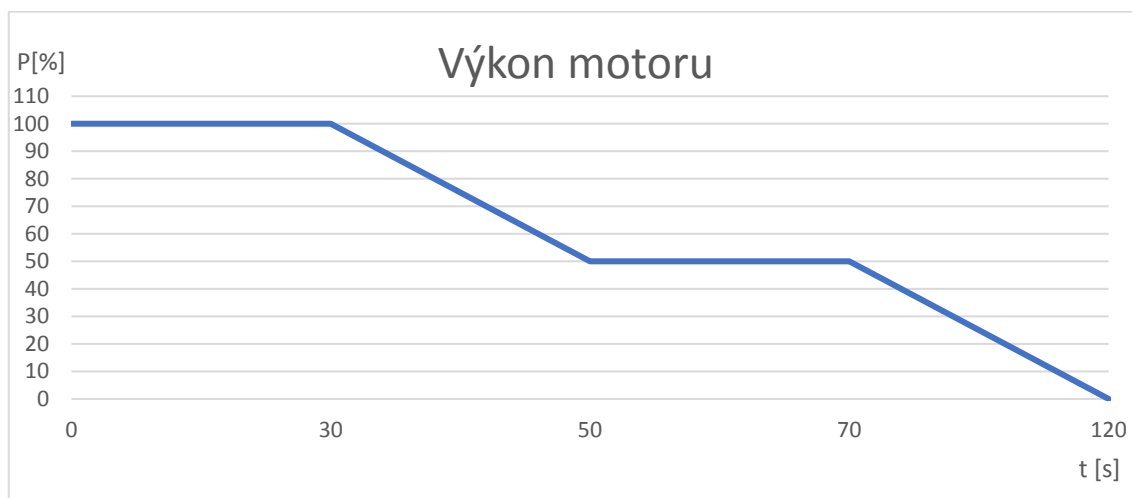


A.2 Letový scénář OUTDOOR

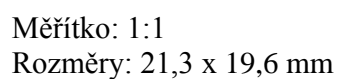
Zmáčknutí
startovacíh
o tlačítka
TL.

➤

t [s]	Výkon motoru [%]	SERVO1 VOP [%]	SERVO2 SOP [%]
0	100	0	0
30	100	0	0
50	50	20	30
70	50	20	30
120	0	20	50

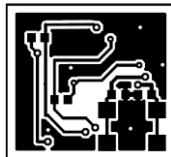


B.1 Schéma zapojení modulu časovače



B.3 Deska plošného spoje časovače – bottom (strana spojů)

✚ Ondrej Fiser BOTTOM ✚



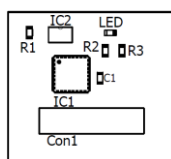
✚

Měřítko: 1:1

Rozměry: 21,3 x 19,6 mm

B.4 Osazovací výkres desky plošného spoje časovače – top (strana součástek)

✚



✚

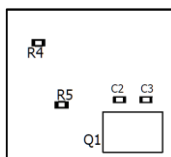
✚

Měřítko: 1:1

Rozměry: 21,3 x 19,6 mm

B.5 Osazovací výkres desky plošného spoje časovače – bottom (strana spojů)

✚



✚

✚

Měřítko: 1:1

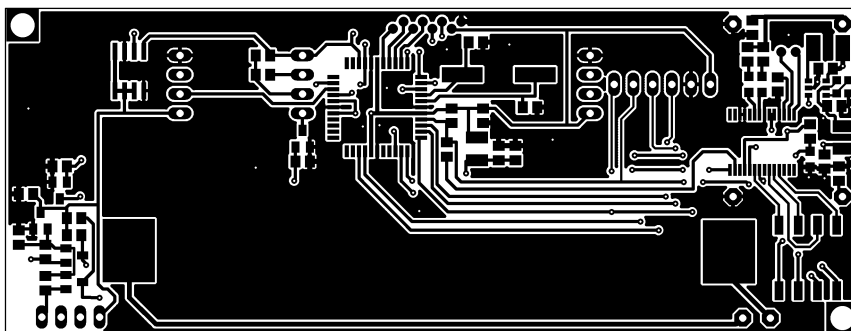
Rozměry: 21,3 x 19,6 mm

C.1 Schéma zapojení procesorové desky programovacího modulu



C.2 Deska plošného spoje procesorové desky programovacího modulu – top (strana spojů)

Ondrej Fiser TOP

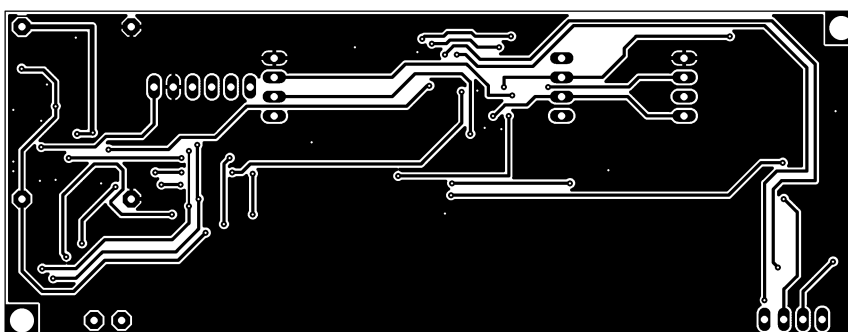


Měřítko: 1:1

Rozměry: 112,7 x 43,0 mm

C.3 Deska plošného spoje procesorové desky programovacího modulu – bottom (strana spojů)

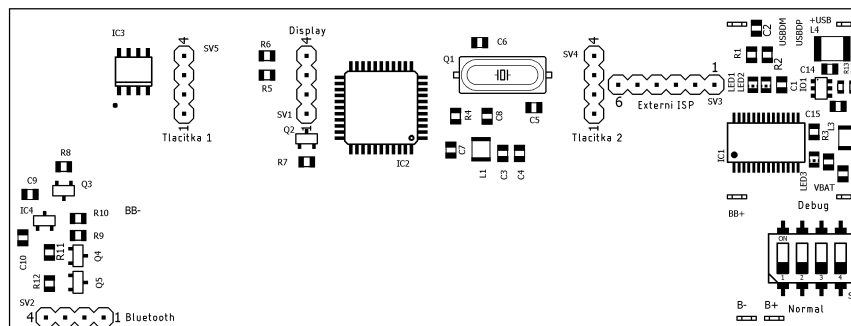
Ondrej Fiser BOTTOM



Měřítko: 1:1

Rozměry: 112,7 x 43,0 mm

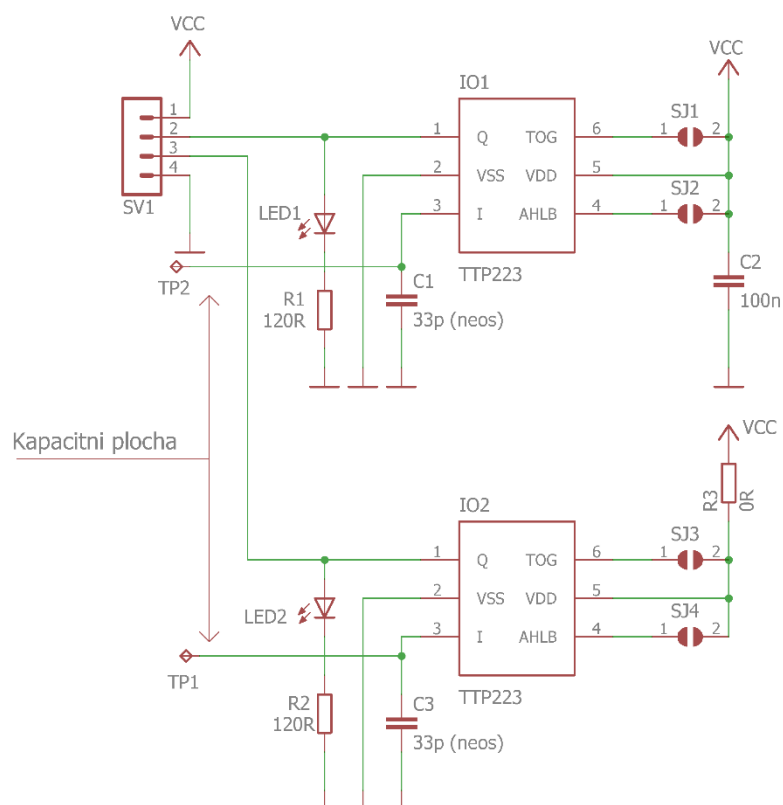
C.4 Osazovací výkres desky plošného spoje procesorové desky programovacího modulu – top (strana součástek)



Měřítko: 1:1

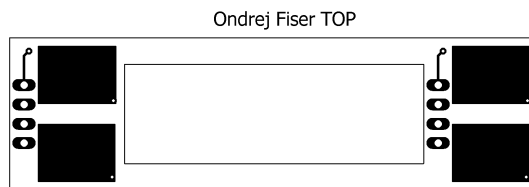
Rozměry: 112,7 x 43,0 mm

C.5 Schéma zapojení kapacitních tlačítek programovacího modulu



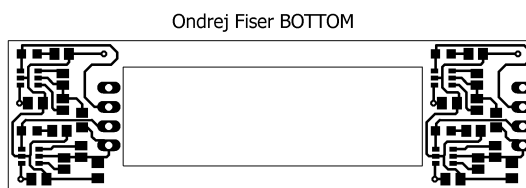
Pozn.: Uvedené schéma je pouze pro jeden pár tlačítek.

C.6 Deska plošného spoje tlačítek programovacího modulu – top (strana spojů)



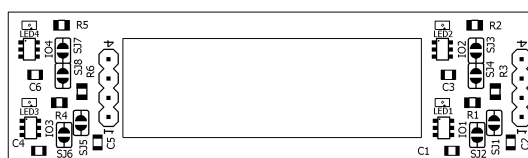
Měřítko: 1:1
Rozměry: 70 x 20 mm

C.7 Deska plošného spoje tlačítek programovacího modulu – bottom (strana spojů)



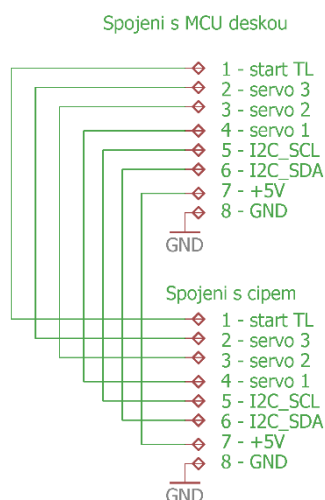
Měřítko: 1:1
Rozměry: 70 x 20 mm

C.8 Osazovací výkres desky plošného spoje tlačítek programovacího modulu – bottom (strana spojů)

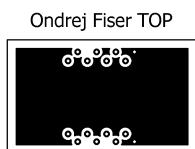


Měřítko: 1:1
Rozměry: 70 x 20 mm

C.9 Schéma zapojení propojovací desky programovacího modulu

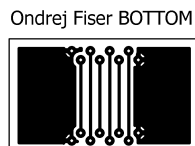


C.10 Deska plošného spoje propojovací desky programovacího modulu – top (strana spojů)



Měřítko: 1:1
Rozměry: 25 x 15 mm

C.11 Deska plošného spoje propojovací desky programovacího modulu – bottom (strana spojů)



Měřítko: 1:1
Rozměry: 25 x 15 mm

D SEZNAMY POUŽITÝCH SOUČÁSTEK

D.1 Seznam součástek modulu časovače

Označení součástky	Počet	Hodnota	Typ součástky	Pouzdro
Q1	1	16 MHz	Krystal	12SMX
C1	1	100 nF/50 V	Kondenzátor	0603
C2, C3	2	16 pF/50 V	Kondenzátor	0603
R1	1	10 kΩ/0,1 W	Rezistor	0603
R2, R3	2	500 Ω/0,1 W	Rezistor	0603
IC1	1	ATMEGA8A-MU	Mikrokontroler	VQFN32
IC2	1	24LC64T-I/MC	EEPROM paměť	DFN8
LED	1	HSMF-C165	Dvojitá LED dioda	0603
Con1	1	NX1251-10SMR	Konektor	-

D.2 Seznam součástek programovacího modulu

Seznam součástek desky tlačítek				
Označení součástky	Počet	Hodnota	Typ součástky	Pouzdro
LED1, LED2, LED3, LED4	4	OSG50805C1E	Zelená LED dioda	0805
C2, C5	2	100 nF/50 V	Kondenzátor	0805
C1, C3, C4, C6	4	33pF/50 V (neos)	Kondenzátor	0805
R1, R2, R4, R5	4	1 kΩ/0,125 W	Rezistor	0805
R3, R6	2	0 Ω/0,125 W	Rezistor	0805
IO1, IO2, IO3, IO4	4	TTP223	Dotykový senzor	SOT23-6
SV1, SV2	2	ZL262-4SG	Zásuvka do DPS	-

Seznam součástek procesorové desky				
Označení součástky	Počet	Hodnota	Typ součástky	Pouzdro
Q1	1	16 MHz	Krystal	HC49
Q2, Q3, Q4, Q5	4	BSS123	Tranzistor	SOT23
C1, C2, C3, C4, C7, C8, C14	7	100 nF/50 V	Kondenzátor	0805
C5, C6	2	18 pF/50 V	Kondenzátor	0805
C9, C10	2	1 uF/50 V	Kondenzátor	0805
C11, C13, C15	3	10 uF/10 V	Kondenzátor	0805
R1, R2, R3	3	1 kΩ/0,125 W	Rezistor	0805

Seznam součástek procesorové desky – pokračování

R4, R5, R6, R7, R8, R9, R10, R11, R12	9	10 kΩ/0,125 W	Rezistor	0805
R16	1	309 kΩ/0,1 W	Rezistor	0603
R13	1	976 kΩ/0,1 W	Rezistor	0603
IC1	1	FT232RL	Převodník USB/UART	SSOP28
IC2	1	ATMEGA644P	Mikrokontroler	TQFP44
IC3	1	24LC512	EEPROM paměť	SO-08
IC4	1	TS9011SCX	Stabilizátor nap.	SOT23
IO1	1	MCP1640	Boost měnič	SOT23-6
L1, L3	2	NLV32T-4R7J-PF	Tlumivka	1210
L4	1	LQH43CN4R7M03L	Tlumivka	1812
LED1, LED2, LED3	3	OSG50805C1E	Zelená LED dioda	0805
SV1, SV2, SV4, SV5	4	ZL2029-20	Patrová lišta	-
SV3	1	ZL211-20KG-S	Úhlová lišta	-
S1	1	ESD104E	Dip switch	-

Další komponenty do sestavy				
Označení součástky	Počet	Hodnota	Typ součástky	Pouzdro
-	1	OLED I2C Displej 128x32 0,91"	OLED displej	-
-	1	KEYS1043	Klips na baterii	-
-	1	S1501	Posuvný přepínač	-
-	1	03962A	Bluetooth modul	-
-	1	HC-05	Nabíjecí modul	-
-	1	ACCU-NCR18650B	Li-ion baterie	-

E TECHNICKÉ PARAMETRY

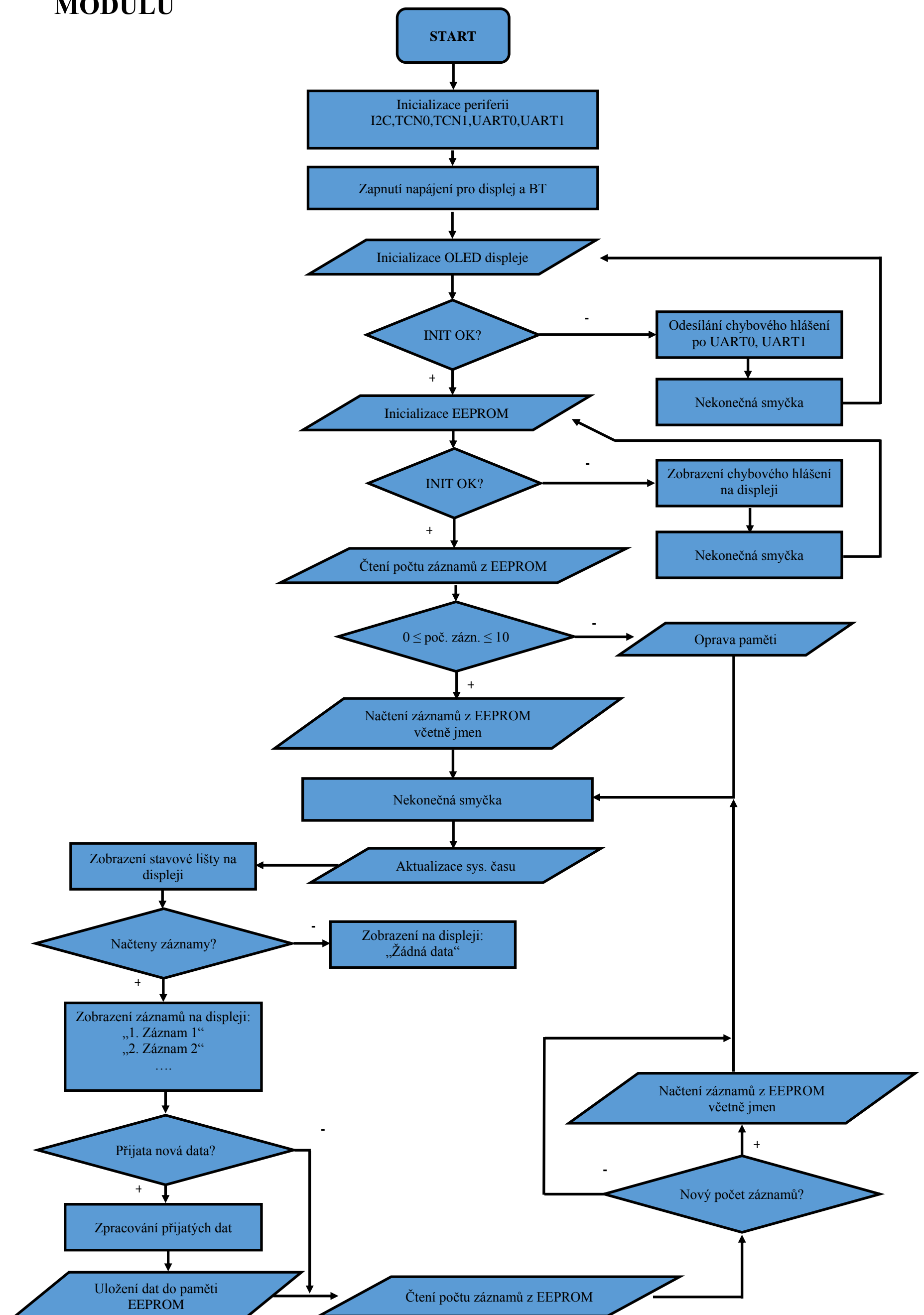
E.1 Časovač pro RC modely

	Procesor	Paměť EEPROM	Celý modul
Napájecí napětí [V]	2,7-5,5	2,5-5,5	2,7-5,5
Max. Spotřeba ($U_{\text{nap}}=5\text{ V}$) [mA]	11	3	13
Velikost paměti [kB]	8		
Nejmenší možný časový krok výstupu (střídy) [μs]	10		
Nejmenší úhel natočení serva [$^{\circ}$]	0,9		
Počet nezávislých výstupů	3		
Nejmenší časové rozlišení (nejkratší doba, kterou může trvat jeden záznam) [ms]	100		
Maximální počet uložených dat časových záznamů	1998		
Maximální délka všech záznamů [min]	109		
Programovací rozhraní paměti	I2C		
Programovací rozhraní procesoru	ISP		

E.2 Programovací modul

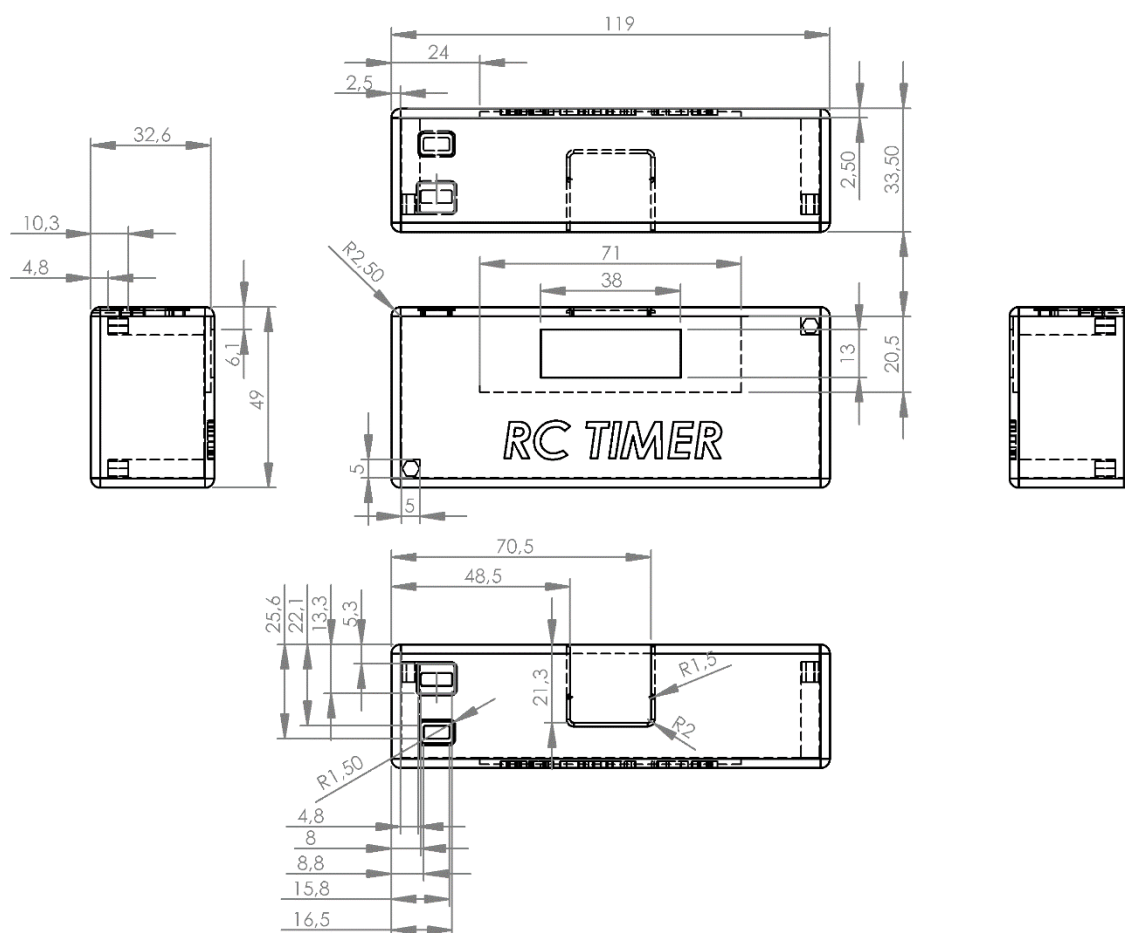
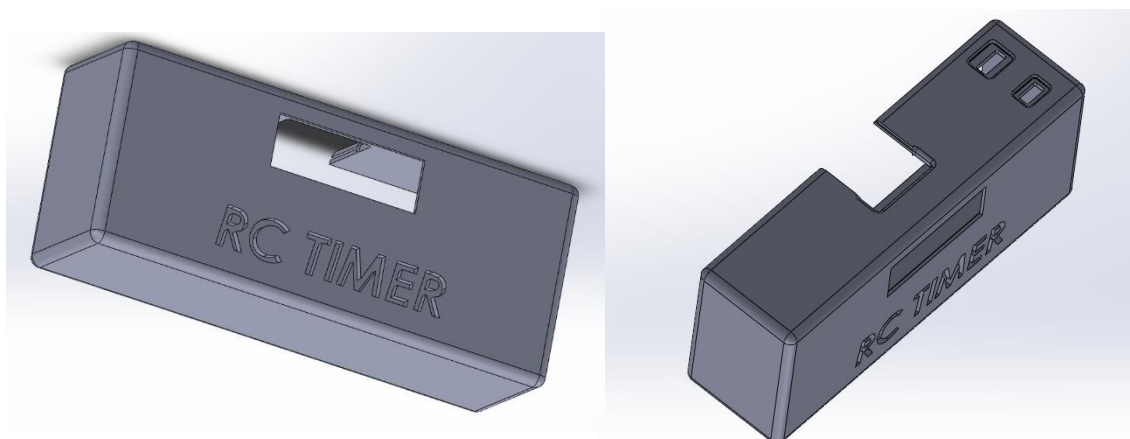
	Celý modul
Pracovní napájecí napětí [V]	2,5 - 6
Max. Spotřeba (poh. stav, BT-zapnuto, nepřipojeno) [mA]	93
Max. Spotřeba (poh. stav, BT-zapnuto, připojeno) [mA]	60
Max. Spotřeba (poh. stav, BT-vypnuto) [mA]	52
Max. Spotřeba (spankový režim) [mA]	13
Min. výdrž na jedno nabití [h]	36
Velikost paměti [kB]	64
Počet uložených sekvencí dat	10
Maximální počet uložených časových záznamů	15962
Komunikační rychlost po UARTu [Bd]	9600
Programovací rozhraní	ISP, USB

F VÝVOJOVÝ DIAGRAM INICIALIZACE PROGRAMOVACÍHO MODULU

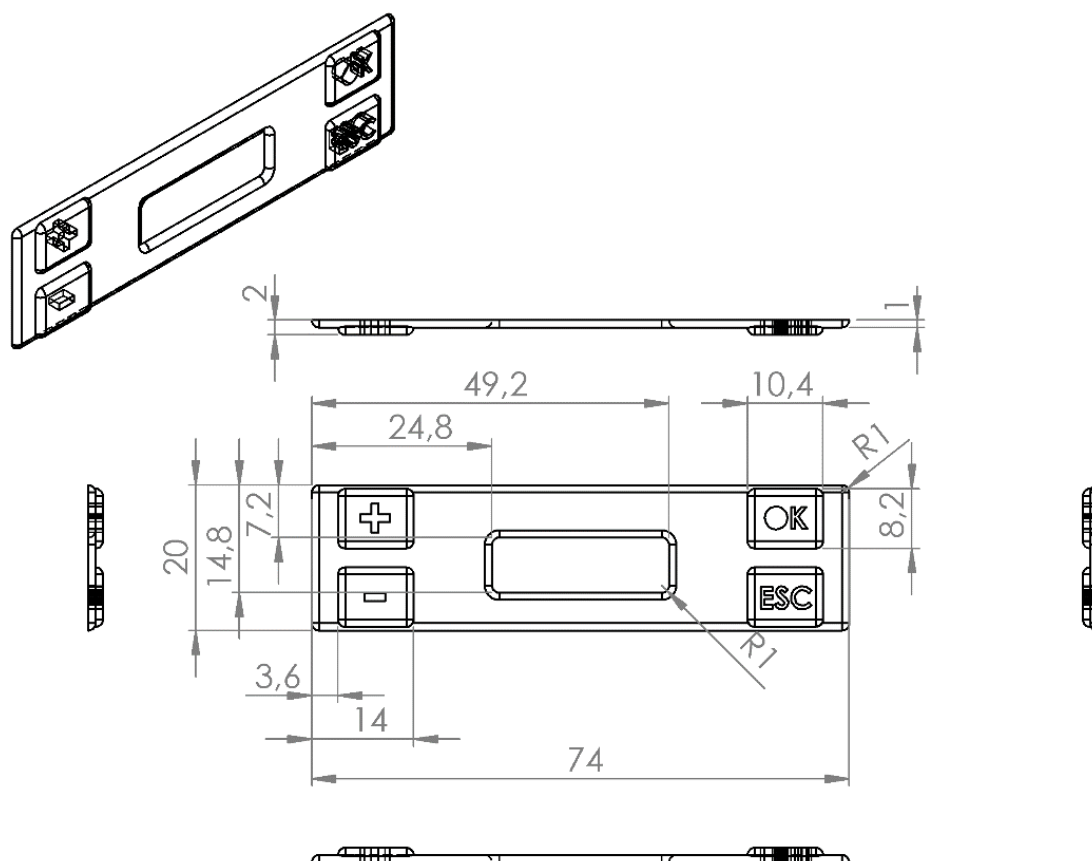
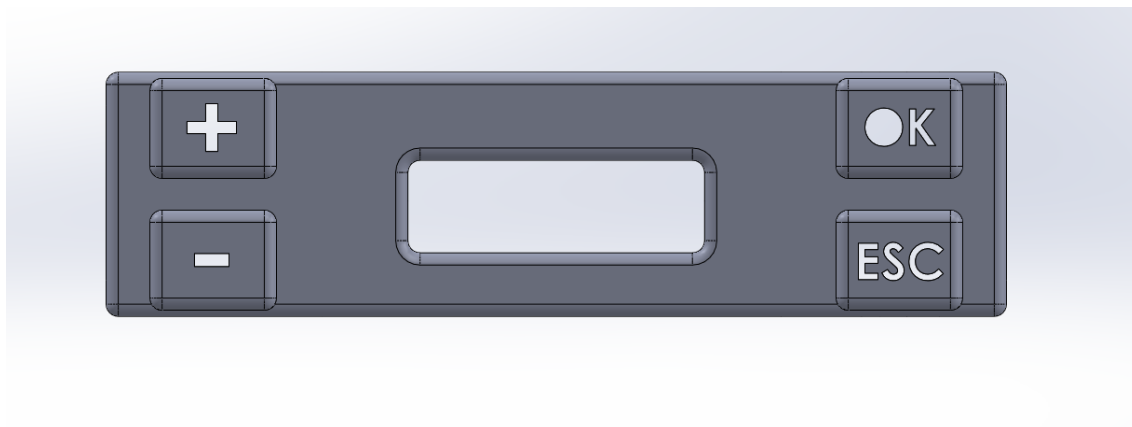


G VÝKRESOVÁ DOKUMENTACE ÚLOŽNÉHO BOXU

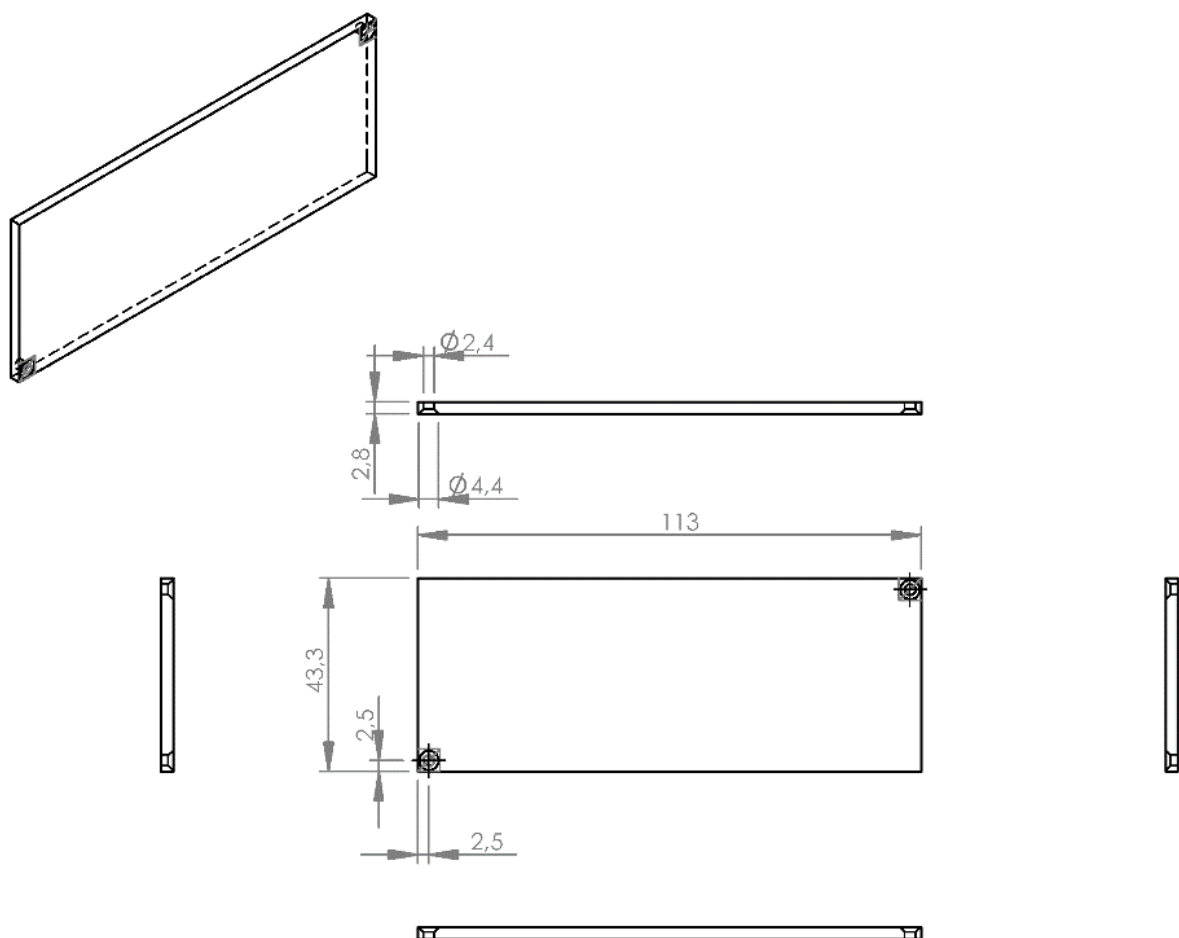
G.1 Úložný box – hlavní část



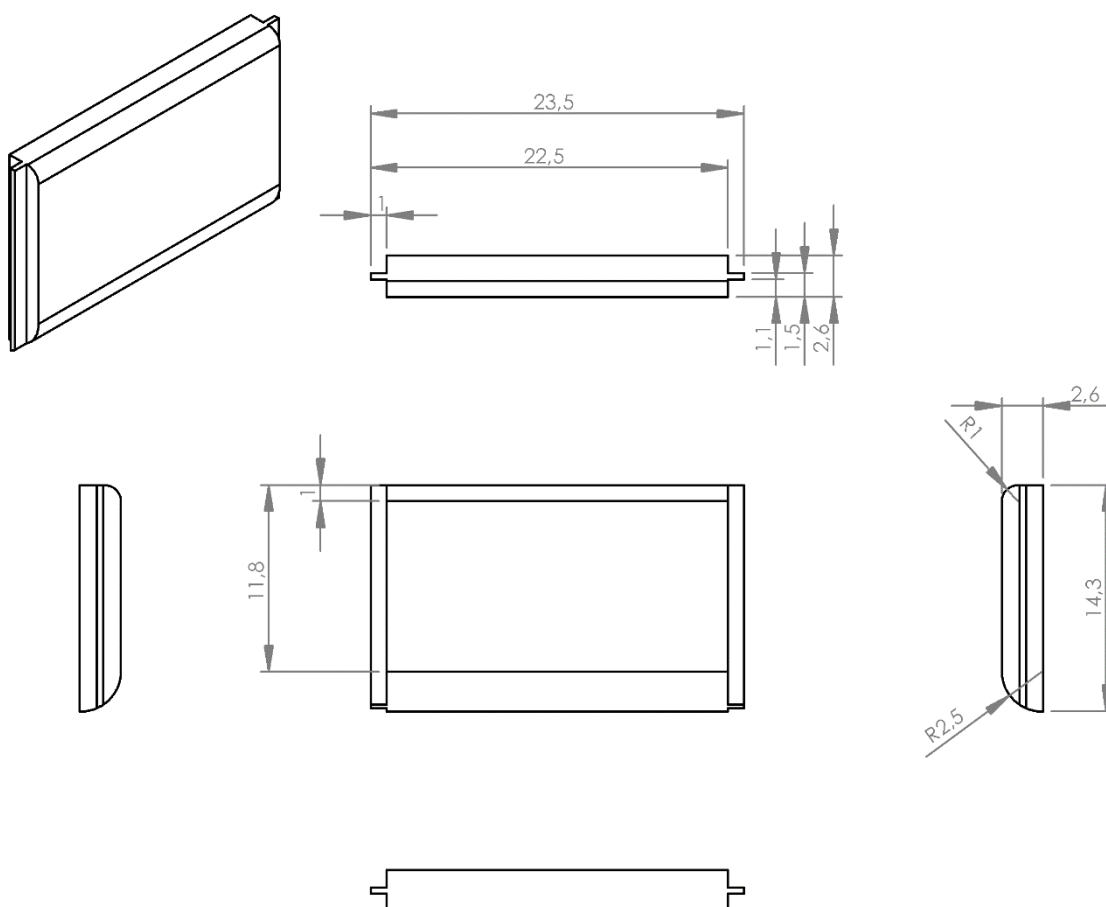
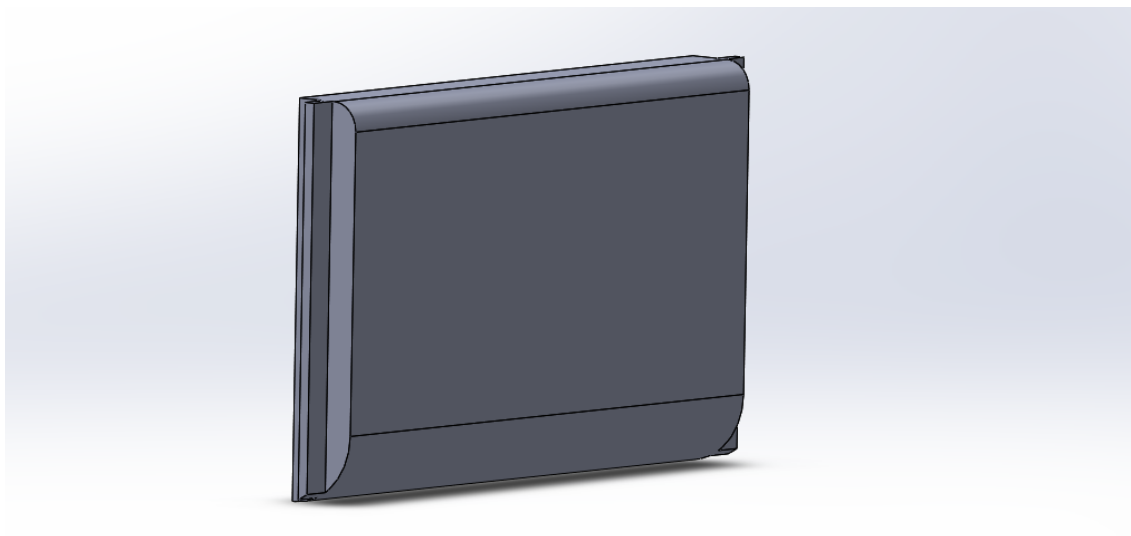
G.2 Úložný box – panel



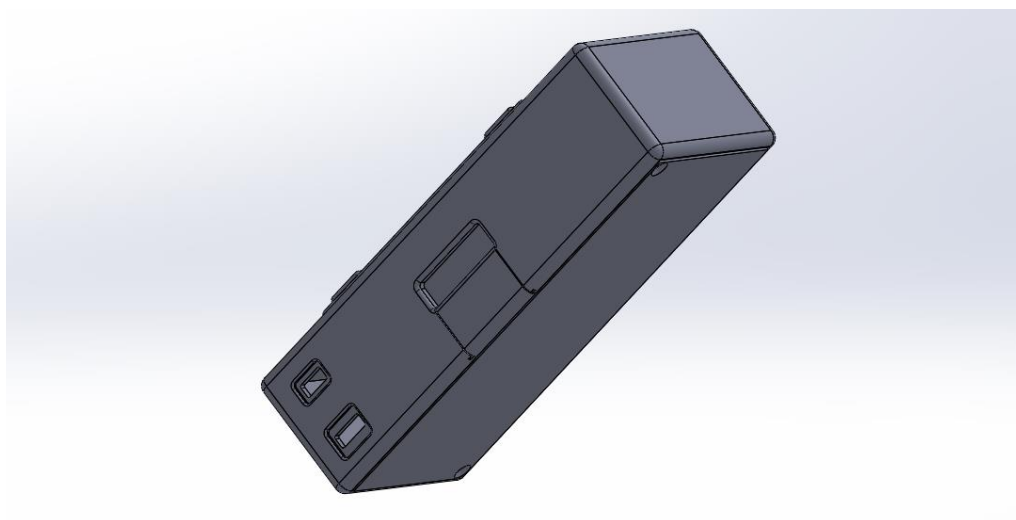
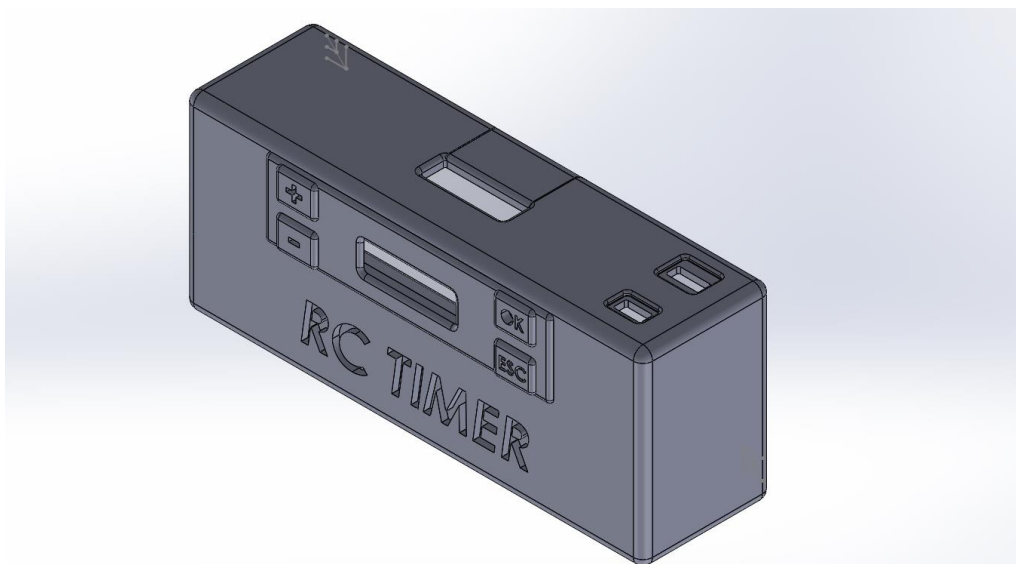
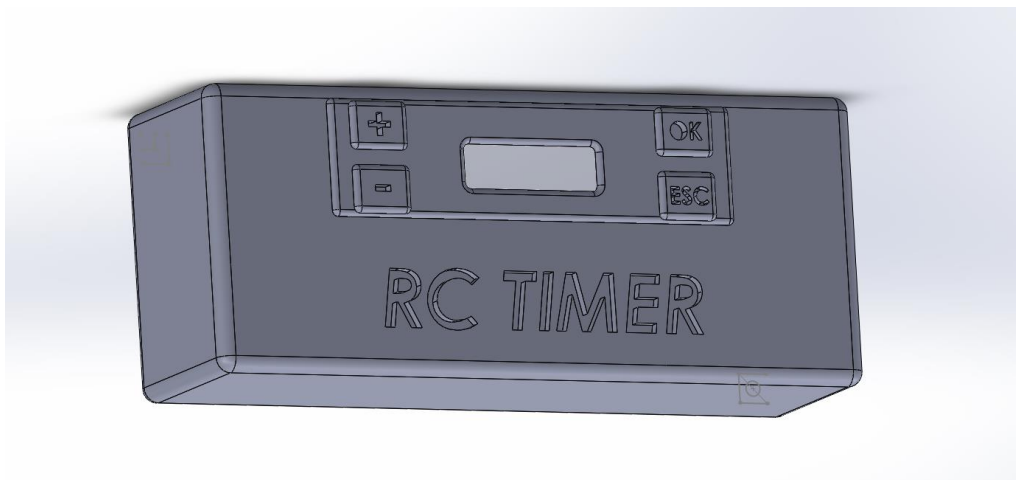
G.3 Úložný box – zadní kryt



G.4 Úložný box – záslepka



G.5 Úložný box – sestava



OBSAH PŘILOŽENÉHO CD

- text diplomové práce
- výkresová dokumentace všech DPS ve formátu pro Eagle 8.3.2
- výkresová dokumentace pro výrobu DPS ve formátu PDF, včetně osazovacích plánů
- seznamy součástek
- výkresová dokumentace pro výrobu úložné krabičky ve formátu pro SolidWorks 2017
- projekt včetně všech okomentovaných zdrojových kódů firmware modulu časovače ve formátu pro Atmel Studio 7.0
- projekt včetně všech okomentovaných zdrojových kódů firmware programovacího modulu ve formátu pro Atmel Studio 7.0
- projekt aplikace „Terminal“ pro MS Windows ve formátu pro Visual Studio 2015
- zkompileovaná aplikace zabalená do instalačního balíku, včetně MS Access Database Engine
- projekt aplikace „Bluetooth Control app“ pro Android ve formátu pro Android Studio 3.1.1
- program „AVRDUDE-GUI“ pro update firmware programovacího modulu
- program „AVRDUDESS“ (včetně ovladačů pro programátor USB-ASP) pro update firmware modulu časovače